Titre: Title:	A comparison of two probabilistic network approaches in the domain of knowledge assessment
Auteur: Author:	Peyman Meshkinfam
Date:	2005
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Meshkinfam, P. (2005). A comparison of two probabilistic network approaches in the domain of knowledge assessment [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7646/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/7646/
Directeurs de recherche: Advisors:	
Programme: Program:	Non spécifié

UNIVERSITÉ DE MONTRÉAL

A COMPARISON OF TWO PROBABILISTIC NETWORK APPROACHES IN THE DOMAIN OF KNOWLEDGE ASSESSMENT

PEYMAN MESHKINFAM
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
DECEMBER 2005

Your file Votre référence
ISBN: 978-0-494-16816-5
Our file Notre référence
ISBN: 978-0-494-16816-5

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

A COMPARISON OF TWO PROBABILISTIC NETWORK APPROACHES IN THE DOMAIN OF KNOWLEDGE ASSESSMENT

présenté par : MESHKINFAM Peyman

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. GALINIER Philippe, Doct., président

M. DESMARAIS Michel, Ph.D., membre et directeur de recherche

M. BOUDREAULT Yves, Ph.D., membre

To my wife, Maryam, my mother Parvin and
my father Gholamali

ACKNOWLEDGMENT

There are a number of people without whom this work might not have been completed, and to whom I am greatly indebted.

I want to take this opportunity to thank, professor Michel Desmarais, my graduate supervisor, for his constant care and attention and valuable technical advise during my work. Moreover, I would like to thank my wife Maryam Sepehr for her constant support and encouragement as well as her contribution in translation and organization of the thesis.

A very special thank to my mom for nurturing me through my life and her active support of me in my determination to find and realize my potential.

I also want to thank Xiaoming Pu, Alejandro Villarreal Morales, Tamer Rafla for their support and help.

I am grateful to Jiri Vomlel for providing the data used in this study.

ABSTRACT

Student models are fundamental elements of intelligent, adaptive learning environments. Constructing a student models, requires a choice between using a handcrafted fine-grained model, which provides precise cognitive diagnostics, or a more general statistical framework that is more data driven but lacks precise diagnostic in the case of small training data. Fine-grained model such as Bayesian Networks, have always been prevalent in the Intelligent Tutoring field. They are considered highly powerful modeling and inferencing techniques because they make few assumptions and they can represent complex relationships among variables with efficiency and parsimony. In comparison the statistical data driven approach is more typical of the psychometric field.

This study is part of a wider research program to better assess and compare the strength and weaknesses of these approaches. It focuses on the comparison of Vomlel's (2004) Bayesian Network model of basic arithmetic skills, to a simple Bayes posterior probability update approach under strong independence assumptions, named POKS (Desmarais, Maluf, and Liu 1995). The models are compared on the ground of their predictive accuracy.

The experimental simulations are based on data from 149 pupils who have participated in an arithmetic test. The result of a 20 question test for these pupils (provided by Vomlel) is used as input for this simulation. In both cases, the simulation consists in choosing the most informative item based on the information gain criteria, and feeding the actual answer to the knowledge assessment technique (BN and POKS). The result is then compared with the actual answers and the process is repeated from the first to the last test item and for each subject.

The results show that both approaches can classify examinees as master or non-master effectively and efficiently. The simulation results shows that item-to-item structures provide higher predictive power at the item level than the Bayesian network approach,

but lower predictive accuracy for concepts. Another experiment combining the two approaches provides further insights. In this experiment the observed items are first processed by POKS. The initial set of observed items is thereby augmented by POKS inferences before it is fed to the BN.

The combination of approaches does yield better performance that each individual one for predicting items. However, no improvement can be achieved at the concept prediction level. The experiment also shows the strong influences of the item selection strategy namely whether items only or items plus concepts are included to choose optimal item.

Both simulations provide evidence that POKS does add information to a BN organized in a hierarchy of concepts with items as leaf nodes. Item to item knowledge structures do seem to provide information that is not modeled by such BN. On the other hand, the BN in this experiment does provide better concept assessment. That confirms the ability of a BN to model complex, non-linear relationships among concepts and items.

As intelligent learning environments evolve and become more popular, knowledge assessment models and techniques will become more pervasive. The availability of simple and automated techniques that are both effective and efficient, relying on little data and allowing seamless updates of test content, will be critical to their success in commercial applications.

Keywords: Bayesian networks, graphical models, CAT, adaptive testing, student models, POKS, Bayesian Network

SOMMAIRE

Introduction

Les environnements d'apprentissage intelligents sont conçus pour s'adapter à la connaissance et aux besoins individuels des étudiants. Ils modélisent la compréhension de l'étudiant sur un sujet et utilisent cette information pour adapter les instructions et le contenu d'apprentissage présenté.

La littérature fournit divers exemples d'environnements d'apprentissage intelligents comme : les environnements de cours personnalisés (Mayo *et al.* 2001, VanLehn *et al.* 2005), l'hypertexte adaptatif et les manuels intelligents (Schwarz *et al.* 1996), les guides d'apprentissage de l'étudiant (Khuwaja, Desmarais, et Cheng 1996) et l'hyper-manuel adaptatif (Brusilovsky, Schwarz, et Weber 1996, Brusilovsky, Eklund, et Schwarz 1997). Ces systèmes tentent de saisir la connaissance de l'étudiant sur le sujet et utilisent cette information pour déterminer le niveau de difficulté approprié du matériel présenté.

Le test adaptatif (TA, ou « Computer Adaptive Testing » en anglais) peut lui-aussi être considéré comme un environnement d'apprentissage intelligent, parce qu'il emploie un modèle d'utilisateur pour adapter des items présentés à l'étudiant. Il peut aussi être considéré comme un outil pour un tel environnement parce qu'il peut aider à établir un modèle d'étudiant en inférant les connaissance et compétences maîtrisées à partir des réponses précédentes.

En adaptant un test à un étudiant particulier, nous cherchons une évaluation de la compétence d'un étudiant avec un minimum d'épreuves. L'ordinateur peut mettre à jour l'évaluation des habiletés du candidat après l'observation de chaque item et, en retour, cette évaluation peut être employée dans le choix des items successifs. Avec une banque appropriée d'items et une variance élevée d'habiletés du candidat, le TA peut être beaucoup plus efficace qu'un essai traditionnel de papier-et-crayon pour lequel la

séquence d'items est fixe. Avec le TA, le choix des items permet d'optimiser la quantité d'information obtenue en fonction du niveau d'habileté estimé et de réduire ainsi le nombre d'items administrés.

Bien que le TA ait été traditionnellement employé pour classifier l'étudiant comme un maître ou un non-maître du sujet, il peut également être employé pour diagnostiquer les qualifications et lacunes (fausses conceptions) d'un étudiant à un niveau plus détaillé. En effet, pour les environnements d'apprentissage, les modèles d'étudiant doivent fournir l'évaluation des concepts détaillés maîtrisés de l'utilisateur et même parfois de fausses conceptions. Une simple classification s'avère généralement insuffisante. Des réseaux bayésiens (BN) et d'autres approches probabilistes sont souvent employées pour obtenir une telle évaluation détaillée. Nous étudions la façon d'établir des modèles d'étudiant précis dans un cadre de TA avec les modèles graphiques.

Nous comparons deux de ces modèles, le modèle des structures d'ordre partiel de la connaissance (POKS), basé sur des structures par paires d'items entre eux qui peuvent être apprises à partir de petits échantillons de données, et le modèle de réseau bayésien (BN). Bien qu'ils soient tous deux des modèles probabilistes, ils comportent des différences significatives quant à leurs hypothèses, au travail préalable d'ingénierie de connaissance et finalement quant à la quantité de données nécessaires à l'étape d'apprentissage de la structure ou des paramètres. Nous comparons un modèle de réseau bayésien (BN) à un modèle simple basé sur l'application de la règle de Bayes avec l'hypothèse d'indépendance locale (POKS). Le domaine modélisé est celui de la maîtrise des opérations arithmétiques de base avec des fractions.

Ordres partiels de connaissances (Partial Order Knowledge Structures, POKS)

Les structures d'ordre partiel de la connaissance (Partial Order Knowledge Structure, POKS, Desmarais *et al.* 1996) constituent une approche de modélisation bayésienne qui

se base sur la théorie de l'espace de la connaissance (Doignon et Falmagne 1999). Cette théorie stipule que nous apprenons à maîtriser des qualifications dans un ordre donné (partiel). Le cadre de POKS repose sur plusieurs hypothèses fortes pour réduire la complexité dans la modélisation bayésienne. L'approche POKS permet une modélisation bayésienne de la structure de la connaissance, en créant des liens entre les items d'un test eux-mêmes (question-par-question), en accord avec la théorie des espaces de connaissances. Contrairement à la plupart des modèles basés sur un réseau bayésien qui fournissent les structures contenant des nœuds représentants à la fois des concepts et des items, le réseau général du POKS est défini exclusivement avec des items de test et aucun concept n'est inclus.

L'approche POKS effectue une propagation d'évidence dans une structure afin d'inférer l'état de la connaissance d'un individu. L'induction de telles structures est basée sur trois tests statistiques portant sur des probabilités conditionnelles et sur l'indépendance conditionnelle des items de la connaissance. Des résultats expérimentaux avec cette approche ont prouvé que la technique réussit à inférer l'état de la connaissance d'un individu, soit par l'observation des actions d'un utilisateur (Desmarais, Giroux, et Larochelle 1993) ou par une série de questions-réponses (Desmarais, Maluf et Liu 1995, Desmarais et Pu 2005). Contrairement aux réseaux bayésiens, l'approche POKS n'a besoin d'aucune étape d'ingénierie de connaissance pour construire le réseau ce qui lui confère un attrait pratique important.

Fondée sur l'hypothèse de l'indépendance locale, la procédure de construction du réseau consiste à comparer par paire les items pour déterminer l'existence d'une relation. Pour établir une relation entre deux nœuds dans la structure du réseau, l'approche POKS emploie trois tests statistiques. Deux tests permettent de valider si les probabilités conditionnelles P($X_b \mid X_a$) et P($\sim X_a \mid \sim X_b$) sont au-dessus d'un seuil fixé et ils se basent sur une distribution binomiale. Un dernier test vérifie l'indépendance et il est basé sur le test de χ^2 effectué sur la table de contingence de deux items. Deux paramètres utilisés

lors de ces tests, un seuil de probabilité conditionnelle et l'erreur alpha du test χ^2, indiquent la force des relations de conjecture et l'erreur d'inférence tolérée.

Une fois que la topologie du POKS est induite à partir de données, elle peut être employée pour l'inférence de la connaissance.

En résumé, les caractéristiques principales du POKS sont les suivantes :

- POKS définit un algorithme pour l'induction, à partir de données empiriques, de la structure des relations entre les items.
- POKS fait l'hypothèse de l'indépendance locale entre les relations. Le modèle fait essentiellement l'hypothèse que nous pouvons limiter seulement la construction d'un réseau à des relations binaires (ce qui permet l'induction du réseau avec un très petit nombre de cas de données par rapport à un modèle qui doit composer avec des relations tertiaires ou plus).

- Le réseau POKS est seulement défini sur des items de tests et aucun nœud de concepts n'est inclus. En imposant cette règle, on évite le besoin d'ingénierie de connaissance pour construire le réseau.

Dans POKS, chaque item représente un nœud. On assigne à chaque nœud, X_i, une probabilité qui représente les chances de succès d'un candidat à cet item, $P(X_i)$. L'ensemble de toutes ces probabilités représente le modèle de connaissance de l'étudiant.

Les relations dans POKS (comme des relations de conjecture dans la théorie de l'espace de la connaissance) indiquent l'ordre (partiel) dans lequel les gens apprennent à maîtriser des items de la connaissance. Les relations de conjecture sont analogues aux relations de causalité trouvées dans les réseaux bayésiens et permettent le même type d'inférences. Par exemple, une relation $X_a \Rightarrow X_b$, signifie qu'en observant un étudiant ayant un succès pour l'item X_a, la probabilité estimée du succès de l'item X_b augmentera. En revanche, un échec pour l'item X_b diminuera la probabilité estimée du succès de l'item X_a.

La mise à jour de la probabilité d'un item de la maîtrise dans le réseau se produit par l'observation d'items réussis ou échoués. L'algorithme pour la propagation d'évidence est essentiellement basé sur le calcul de probabilité postérieure selon le cadre bayésien classique, mais dans sa version basée sur des ratios de chance (odds ratios). Ce cadre emploie deux rapports de chance : la vraisemblance de la suffisance (Likelihood of Sufficiency, LS) et la vraisemblance de la nécessité (Likelihood of Necessity, LN). En utilisant l'hypothèse de l'indépendance locale, les rapports de chance sont combinés comme le produit du LS de chaque parent observé.

Évaluation de la connaissance dans un réseau bayésien

Le réseau bayésien (BN) est un cadre théorique bien établi dans le raisonnement probabiliste (Neapolitan 2004). Il a été employé dans la dernière décennie pour établir des modèles d'utilisateur et, plus récemment, pour le TA.

Afin de concevoir un test d'évaluation et diagnostiquer la présence ou l'absence des qualifications d'une personne, un concepteur d'essai éducatif indique un ensemble de qualifications, d'habiletés, de fausses conceptions, etc., et une banque de questions, de tâches, etc. Chaque compétence et chaque question sont représentées par une variable aléatoire ayant des ensembles finis de valeurs. Différents modèles ont été proposés pour construire le modèle d'étudiant basé sur le BN (Conati *et al.* 2002, Millan *et al.* 2000). Ici, nous choisissons une approche de mise en application par Vomlel (2004), qui est inspirée d'Almond et Mislevy (1999). L'approche présentée par Vomlel décrit un algorithme pour construire un modèle étudiant dans le domaine des opérations de base avec des fractions.

Tout comme pour POKS, Vomlel vise à prédire la maîtrise de compétences à partir de l'évidence provenant des réponses de l'étudiant et la structure du réseau peut être apprise à partir de données empiriques. Cependant, la construction d'un réseau bayésien à partir de données est un processus qui peut nécessiter de grandes quantités de données. Le principe général consiste à trouver la topologie la plus probable du réseau étant donné les données observées. Ce processus est basé sur un calcul de vraisemblance, plus

spécifiquement le logarithme de la vraisemblance conditionnelle (CLL, C*onditional Log-Likelihood*). Un problème fondamental est qu'afin de maximiser la CLL pour la structure modèle étant donné une série d'observations, il est nécessaire de rechercher des paramètres modèles sur l'espace entier. Ceci rend la méthode informatique coûteuse en calcul et en données.

Les expériences de Cheng et Greiner (1999) suggèrent que des algorithmes d'apprentissage basés sur la série d'essais de l'indépendance conditionnelle fournissent des classificateurs de réseau bayésien qui performent bien. Vomlel a employé un algorithme d'apprentissage basé sur la contrainte – l'algorithme de PC de Hugin – une variante de l'algorithme de PC original de Spirtes *et al.* (1993) pour apprendre la structure du modèle d'étudiant. Cet algorithme est basé sur la série de tests de l'indépendance conditionnelle.

La structure produite par algorithme de PC dérivent souvent trop de rapports de l'indépendance conditionnelle surtout avec un petit échantillon de données. En outre, elle peut dans certains cas, mettre des relations importantes de dépendance. Ainsi, la structure induite doit être inspectée par un expert du domaine. Vomlel a combiné l'algorithme de PC de Hugin appliqué à des données empiriques et il y a apporté des ajustements avec un expert du domaine modélisé. L'approche est donc une combinaison de méthodes algorithmiques pour construire et paramétriser la structure et d'intervention d'ingénierie de la connaissance.

Choix de l'item

Il est important de maximiser des informations sur le candidat diagnostiqué par le choix des items présentés. Dans le TA, l'augmentation d'information correspond à une diminution de l'incertitude sur le sujet de la présence ou de l'absence des qualifications représentées par la distribution de probabilité des qualifications. Le but principal du choix d'item est d'identifier le niveau des habiletés du candidat avec la précision maximale en utilisant le nombre minimum d'items, ce qui signifie de choisir l'item le plus informatif.

Les critères dépendent des propres caractéristiques de l'item aussi bien que de l'évaluation courante du niveau des habiletés du candidat.

Il existe plusieurs méthodes pour des choix de l'item (Eggen 2004). Trois des méthodes bien connues sont l'Information de Fisher, le Gain de l'Information et les mesures Maximum de Discrimination. Parmi eux le Gain de l'Information est utilisé par Vomlel pour les réseaux bayésiens, tandis que l'Information de Fisher est souvent employée par l'approche IRT, couramment utilisée pour le TA.

Le principe de l'approche de Gain de l'Information est de choisir l'item qui maximisera la réduction prévue de l'entropie de l'essai. Cette stratégie est basée sur le choix de l'item qui réduira au minimum l'entropie globale des items du test. Si toutes les probabilités de l'item sont proches de 0 ou de 1, l'entropie sera petite et il y aura peu d'incertitude au sujet des habiletés du candidat. Le choix de l'item le plus informatif correspond donc à celui dont la valeur attendue d'entropie est la plus basse.

Évaluation des concepts à partir des réponses aux items avec un réseau neural à une couche (perceptron)

Comme mentionné, POKS n'inclut pas de concepts dans sa structure. Il faut donc établir un mécanisme pour évaluer la maîtrise des concepts à partir des réponses aux items. L'approche adoptée ici consiste à utiliser une technique de classification conventionnelle que l'on entraîne avec des données. En effet, Vomlel a demandé à des experts d'évaluer de façon indépendante la maîtrise des concepts à partir des réponses aux items. Ce sont en fait les mêmes données qui lui ont servi à construire le réseau bayésien. Il nous devient donc possible d'utiliser ces données pour l'entraînement. Évidemment, ce n'est pas une approche que l'on peut appliquer en pratique, mais elle s'avère nécessaire pour l'étude.

Nous lions des items de question aux concepts en employant une série de réseaux neuronaux simples à chaînage-avant composés d'un seul nœud caché. Les items servent comme de nœuds d'entrée au réseau qui comporte un concept comme un nœud de sortie.

Les réseaux sont basés sur les liens définis dans le réseau bayésien. Les paramètres des réseaux neuronaux sont entraînés avec les données de maîtrise du concept de Vomlel.

Simulation

Le réseau bayésien de Vomlel (2004) est comparé à l'approche POKS par une méthodologie expérimentale de simulation. Les données de simulation sont celles de Vomlel (2004). Elles consistent en des données quant aux résultats de l'examen de 20 questions (items) administrés à 149 étudiants. Le domaine est celui des compétences élémentaires en arithmétiques. En plus des 20 items, les données contiennent aussi l'évaluation de la maîtrise de 20 concepts (incluant 4 fausses conceptions). Des pédagogues ont estimé cette maîtrise en analysant les réponses aux 20 items. Ces données permettent l'entraînement à la fois du BN et de POKS. Cette situation est atypique, puisque généralement nous n'avons pas le luxe d'avoir un estimé indépendant de la maîtrise de concept et d'entraîner un modèle avec de telles données. Ces données s'avèrent toutefois fort utiles pour les fins de cette étude.

Neuf modèles ont été examinés par Vomlel et nous avons utilisé les résultats du modèle le plus performant.

Résultats

Les simulations fournissent une comparaison entre l'approche POKS et une approche de réseau bayésien à deux niveaux:

1. la prévision des réponses aux questions : la capacité de prédire le succès ou l'échec aux items non répondus;
2. la maîtrise des concepts : la capacité de prédire la maîtrise des concepts tels qu'évalués par les experts;

Exactitude des prévisions des questions

Les résultats de simulation démontrent que la technique du POKS peut prédire des réponses aux questions avec une précision supérieure de 1 – 4% à celle du BN. Le gain est plus significatif lorsque le nombre d'items observés augmente en proportion du nombre restant d'items non-observés. D'autre part, les résultats indiquent que l'habileté du BN pour prédire des résultats aux questions n'est apparemment pas meilleure qu'une stratégie simple qui consiste à utiliser une séquence fixe et commune à tous basée sur le choix des questions les plus incertaines d'abord.

Prévision des concepts

La deuxième évaluation mesure l'exactitude des approches pour prédire la maîtrise de concept. Comme nous avons mentionné, la maîtrise des concepts a été évaluée indépendamment par des experts. Pour cette mesure, l'avantage de la performance est inversée en faveur du modèle de BN. L'approche BN atteint rapidement 74% d'estimation correcte après seulement 5 items observés, et elle se stabilise à presque 92%. La performance de POKS est environ 2 – 3% au-dessous de celle du BN entre la deuxième et le $15^{ième}$ item de question, et elle devient plus proche au-delà de cet intervalle. Nous notons également que la performance de POKS est légèrement plus faible (environ 1%) après que tous les 20 items sont administrés, indiquant que le modèle de réseau neural utilisé n'est pas aussi performant que le BN pour prédire les concepts avec toutes les réponses. En somme, une partie de la différence de performance s'explique du fait que le réseau neuronal n'est pas aussi efficace que le réseau bayésien.

Une interprétation vraisemblable du fait que la performance de POKS est en dessous de celle du BN, notamment après toutes les 20 questions, est que le réseau bayésien exploite des interdépendances de concepts pour évaluer le maîtrise par l'étudiant (comme on pouvait déjà le déduire de l'expérience de Vomlel, 2004), alors que l'approche POKS combinée aux réseaux neuronaux ne modélise pas ces relations et se trouve ainsi défavorisée au niveau des concepts. Malgré le fait que POKS prédise mieux les réponses aux questions, on peut présumer que la modélisation des relations entre concepts apporte

encore plus d'information que le gain d'information de POKS au niveau des items eux-mêmes.

Même en tenant compte de l'écart de 1% de performance après tous les 20 items, il faut aussi noter que POKS est 2 – 3% en deçà de la performance du BN pour une bonne part de la simulation entre 0 et 20 items. Cette différence peut s'expliquer par le choix d'items pour POKS qui n'est basé que sur l'entropie calculée à partir des items, alors que le choix pour le BN est basé sur les items et les concepts. De ce fait, le BN est plus susceptible de réduire l'incertitude au niveau des concepts que ne l'est POKS. Cette explication s'applique aussi, à l'inverse cette fois, pour les résultats meilleurs de POKS au niveau des questions.

Combinaison du POKS avec le BN

Une autre expérience a été entreprise afin de vérifier la possibilité d'améliorer la performance des deux méthodes en les combinant. Nous décrivons ci-dessous la méthode pour combiner les approches et les résultats d'une simulation en utilisant cette approche.

Le principe de la combinaison des approches consiste à ajouter aux items observés ceux inférés par POKS, puis à effectuer l'inférence avec le réseau bayésien. POKS agit ainsi comme un filtre qui ajoute des items inférés avant de transmettre l'information au réseau bayésien.

Afin de déterminer qu'un item est considéré maîtrisé selon POKS, un seuil est employé. Chaque item pour lequel la probabilité de la maîtrise de POKS est plus grand que [1 – SEUIL] est considéré maîtrisé, tandis que des items avec une probabilité plus petite que le SEUIL sont considérés non maîtrisés.

Le seuil a été placé à 0.1 pour cette expérience. Trois différentes comparaisons sont rapportées ci-dessous :

- BN-POKS-POKS : la performance de BN augmentée par les inférences de POKS en utilisant l'algorithme de choix de l'item basé sur l'entropie calculée par POKS.

- BN-POKS-BN : semblable à BN-POKS-POKS sauf que l'algorithme de choix de l'item est basé sur l'entropie calculée par le BN. Contrairement au calcul fait par POKS, le BN inclut les concepts et les items dans le calcul.

- BN (Original/reproduit) : la performance du BN, sans les observations augmentées de POKS, pour les résultats originaux et pour nos résultats reproduits de Vomlel. Comme prévu, les deux courbes sont relativement semblables puisque nous employons les mêmes algorithmes génériques, bien que les détails d'implantation puissent expliquer la petite différence.

Pour réaliser cette expérience, il nous faut répliquer la simulation de Vomlel. La réplication de la simulation a été exécutée avec la librairie de réseau bayésien de Ken Murphy (BNT) (http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html). Nous avons employé les mêmes algorithmes que Vomlel (2004) pour l'inférence, notamment l'algorithme de l'arbre de jonction et l'algorithme EM pour la calibration des probabilités conditionnelles du réseau bayésien.

Les résultats indiquent que toutes les conditions sont relativement semblables. Seulement une courbe est différente des autres de manière significative, le BN-POKS-POKS pour les résultats prédictifs aux questions. Cette condition correspond à la condition où des inférences de POKS sont combinées avec les inférences de BN avec l'algorithme de choix d'item de POKS. Ces résultats indiquent que les inférences additionnelles du POKS contribuent à améliorer la performance de la prédiction d'item seulement quand le choix de l'item de question est basé sur la réduction prévue de l'entropie des items de POKS. Les résultats de BN-POKS-POKS sont très proches de ceux de POKS de la simulation précédente, mais ils montrent une amélioration environ de 2% entre 5 et 10 items observés.

Cependant, cette amélioration pour la prédiction aux items n'est pas transférée à la prédiction aux concepts. L'absence d'amélioration pour la prédiction des concepts sur les améliorations de la prédiction de l'item est inattendue. Néanmoins, elle peut être expliquée par le fait que le choix des items par l'algorithme de l'entropie de BN n'est pas optimal pour POKS et que cela résulte au moins d'inférences « augmentées » alimentées au BN. Il peut également impliquer que le choix des items par l'algorithme du POKS et les items inférés par le BN sont redondants avec l'information déjà dérivée au niveau des concepts, bien que cette explication soit spéculative en ce moment.

Discussion et conclusion

Les résultats de cette étude indiquent un avantage clair de chaque approche :

(1) POKS est plus précis pour prédire les réponses aux items;
(2) BN est plus précis pour prédire la maîtrise des concepts.

La performance du POKS pour la prévision des concepts demeure entre 1 – 3% au-dessous de performance de BN. Cependant, l'avantage de la performance est inversé en faveur du POKS pour l'exactitude prédictive de question. POKS semble prédire mieux des résultats observés des réponses aux items, mais cet avantage n'est pas reflété dans la prédiction des concepts.

En revanche, nous notons que l'avantage de prédiction du concept de BN ne se reflète pas dans son habileté à prédire des résultats au niveau des items. Une partie de ces résultats peut être expliquée par le fait que l'algorithme de choix d'item est différent pour le POKS et pour le BN. POKS optimise le gain de l'information au niveau des items tandis que le BN optimise le gain pour les items et les concepts simultanément. Une étude de simulation qui combine des inférences de POKS et du BN fournit d'autres indices. Elle démontre que lorsque la stratégie de choix d'item du POKS est employée, l'exactitude de la prédiction des réponses aux items par le BN s'améliore pour arriver à des niveaux qui sont même légèrement plus élevés qu'avec l'utilisation seule du POKS. Cependant, cette

amélioration ne se transfère pas à la prédiction de concept, qui demeure près du même niveau obtenu sans les inférences de POKS. Une explication partielle pour le transfert non-effectué peut être qu'en employant la stratégie de choix d'item du POKS, basée seulement sur des items, il y a une légère perte de la performance au niveau de la prévision des concepts.

En tenant compte de tous ces résultats, on peut conclure que le POKS ajoute de l'information à un BN organisé dans une hiérarchie des concepts avec des items comme nœuds de feuille. Des structures de la connaissance d'item à item semblent fournir des informations qui ne sont pas modélisées par un tel BN. D'autre part, dans cette expérience le BN fournit une meilleure évaluation de concept. Cela confirme l'habileté d'un BN à modéliser des relations complexes et non linéaires parmi des concepts et des items.

Cependant, dans la pratique, nous n'avons pas la possibilité d'entraîner un BN avec des données indépendantes d'évaluation de concept. L'approche de BN dans cette étude, bien qu'elle soit instructive pour cette recherche, n'est pas applicable dans la plupart des contextes. Non seulement elle exige un effort d'ingénierie de connaissance mais, sans une évaluation indépendante de concept, l'évaluation des probabilités conditionnelles devient très difficile, et il faut employer des techniques simplificatrices telle que des *portes bruitées-OR/AND* (*noisy-and* et *noisy-or gates*). Le problème de l'applicabilité des méthodes de cette étude s'applique également à POKS. Sans l'évaluation indépendante de concept, POKS ne pourrait se reposer sur un réseau neuronal pour l'évaluation de concept, puisque nous devons compter sur ces données pour l'entraînement du réseau.

Pour cette raison, la performance de POKS pour des résultats de prédiction d'item apparaît la meilleure que nous pouvons actuellement espérer d'une approche étudiée.

Pour aller de l'évaluation d'item aux concepts avec une approche étudiée, on doit ainsi se fonder sur un schéma plus simple tel qu'en considérant la maîtrise de chaque concept comme la somme, peut-être pondérée, de la maîtrise de différents items. Néanmoins, ceci

a l'avantage d'être un processus simple et très familier à chaque éducateur qui a dû écrire un test dont les items représentent un ensemble de connaissances à acquérir.

CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF NOTATIONS AND ABBREVIATIONS

$A \Rightarrow B$	Implication relation
Bin(k,n,p)	Binomial distribution
BN	Bayesian Network
CAT	Computer Adaptive Testing
CLL	Conditional log likelihood
$E(X)$	The expected value of X
e_i	event i
G	Direct Acyclic Graph (DAG)
ILE	Intelligent Learning Environments
IRT	Item Response Theory
KU	Knowledge Unit
LN	likelihood of necessity
LS	Likelihood of sufficiency
n	Number of item
$O(X)$	Odds of X
$P(X = 1)$	Probability of the correct response to an item

$P(E\|F)$	The conditional probability of E given F
$P(E)$	Probability function for event E
POKS	Partial Order Knowledge Structure
U	Knowledge Unit (in equations)
$W_{A \Rightarrow B}$	Weight of implication relation
Ω	Sample space
(Ω, P)	Probability space

LIST OF APPENDICES

CHAPTER 1. INTRODUCTION

1.1 The need for Intelligent Learning Environment and Automated Knowledge Assessment

This thesis is at the frontier of two fields, ***Computer Adaptive Testing*** (CAT) and ***Intelligent Learning Environments*** (ILE). The two fields have evolved separately but they are closely related. CAT is an early application of adaptive interfaces that dates back to the 1960's and is nowadays used in many large scale testing services. ILE encompasses a large family of systems that aim to be adaptive and responsive to the learner's individual needs. CAT can be considered a kind of ILE since it adapts the items of a skill assessment test to an individual's previous responses in order to optimize the information gain and reduce test length. CAT can also be considered a component of an ILE since it can help build a student skill profile and allow the system to tailor the learning strategy for that specific profile. We review the basis of CAT and ILE and the issue of building student model.

1.2 Knowledge assessment and Computer Adaptive Testing

The need for a Computer Adaptive Testing (CAT) system in educational assessment seems obvious: Adaptive testing is the process of adapting a test to a particular learner with the goal of assessing a learner's competency with minimum questioning. When an examinee is administered a test via the computer, the computer can update the estimate of the examinee's ability after each item is observed and then that estimate of ability can be used in the selection of subsequent items. With the right item bank and a high examinee ability variance, CAT can be much more efficient than a traditional paper-and-pencil test. Paper-and-pencil tests are typically "fixed-item" tests in which the examinees answer the same questions within a given test booklet. Since everyone takes every item, all examinees are administered items that are either very easy or very difficult for them.

These easy and hard items are likely to provide relatively little information about the examinee's ability level. Consequently, large numbers of items and examinees are needed to obtain a modest degree of precision.

With computer adaptive tests, the examinee's ability level relative to a group of reference can be iteratively estimated during the testing process and items can be selected based on the current ability estimate. Examinees can be given the items that maximize the information (within constraints) about their ability levels from the item responses. Thus, examinees will receive few items that are very easy or very hard for them. This tailored item selection can result in reduced standard errors and greater precision with only a handful of properly selected items.

The adaptation is typically performed by selecting highly informative test items from a large pool of possible items. This helps classifying the learner as either a master or a non-master of the subject. Since certain test items are highly informative for some learners while not so informative for others, the selection is performed using any information currently known about the learner. In most adaptive testing schemes, this information consists of the learner's response to the previous items. After some amount of information is gathered, the adaptive testing, using some stopping conditions, can terminate with a learner classification; this termination is done as soon as possible to decrease the test length. Rather than placing a student on an ability scale, the goal here is to identify the most likely classification for the examinee. This classification can be dichotomous (e.g., master/non--master) or polychotomous (e.g., master/at-risk/non-master) or involve placement on a categorical or interval scale.

CAT can thus be a valuable tool for an ILE. Through its ability to infer the student's level of mastery based on few evidence. However the traditional goal of classifying examinees as master (non master proves too coarse grained for most ILE). We review the issues and requirements for student modeling in ILE in the next section.

1.3 Previous works

Referring to the literature, we can name a few examples of Intelligent Learning Environments such as: intelligent tutoring systems (Mayo *et al.* 2001, VanLehn *et al.* 2005), adaptive hypertext and intelligent textbooks (Schwarz *et al.* 1996), student study guides (Khuwaja, Desmarais, and Cheng 1996), web based adaptive hyper-textbook, hyper-media and course-ware (Brusilovsky, Schwarz, and Weber 1996, Brusilovsky, Eklund, and Schwarz 1997). These attempts were to capture the student's understanding of the subject and then using this information to determine the difficulty of the material.

Student models are one of the fundamental elements of intelligent and, adaptive learning environments. They must provide fine-grained assessment of the user's concepts mastered and, sometimes, even misconceptions. Bayesian networks such as Graphical probabilistic models and other probabilistic approaches are often used to achieve such detailed assessment.

In order to properly qualify these methods, we investigate different student modeling approaches, used by researchers, over a number of dimensions and qualities. These dimensions are discussed below.

1.3.1 Flexibility and expressiveness

Adaptive based learning systems often rely on fine-grained assessment of abilities and misconceptions. Among different approaches graphical probabilistic models are highly suitable for fine-grained cognitive diagnostic, but they do not readily lend themselves to automated learning. On the other hand, a large body of psychometric theory that dwells on classifying an examinee as master or non-master offers a sound statistical framework for performing global assessment.

1.3.2 Cost of model definition

Usually fine-grained models used in Bayesian networks (Vomlel 2004, Conati, Gertner, and VanLehn 2002) require enormous expert modeling effort. This proves them highly costly for many applications. On the contrary, data driven approaches can completely waive the knowledge engineering effort, although they might need additional work to provide finer grained assessment. We will discuss this further in following chapters.

1.3.3 Scalability

The number of concepts/skills and test items that can be modeled in a single system is another factor that truly affects the appropriateness of an approach. For fine-grained student models, this factor is more difficult to assess and must be addressed on a per case basis. For example, in a Bayesian Network where items and concepts are highly interconnected, complexity grows rapidly and can be a significant obstacle to scalability (Vomlel 2004).

1.3.4 Cost of updating

In order to prevent over exposure of the same test items, skill assessment methods often required frequent updating. Moreover, in rapidly evolving domain, such as in technical training, new items and concepts or skills must be introduced on a regular bases. Therefore approaches that reduce the cost of updating the models will be in demand. This issue is closely tied to the knowledge engineering effort and is better addressed in those models which have the ability to be constructed and parameterized with a small data sample (Desmarais, Maluf, Liu 1996)

1.3.5 Accuracy and reliability of prediction

The ability of the model to provide an accurate assessment with the minimum number of questions is a crucial factor in student modeling applications specifically in CAT. Models that can yield confidence intervals, or the degree of uncertainty of their

inferences/assessment, are thus very important in this field as well as in many context in which measures of accuracy is relevant.

1.3.6 Reliability

It is of high importance to investigate the reliability and sensitivity of a model to environmental factors. These factors can be the skills of the knowledge engineer, the robustness to noise in the model and/or in the calibration data. Handcrafted models, in particular, are subject to individual differences and human biases and therefore, they lack necessary means to predict their reliability.

1.3.7 Mathematical foundations

Models relying on sound and rigorous mathematical foundations are by and large considered better candidates over ad hoc models due to their support to assess accuracy and reliability. These robust Models can often be automated using standard numerical modeling techniques and software packages

1.3.8 Approximations, assumptions, and hypothesis

Almost all models, in order to be applicable to a specific context, take advantage of a number of simplifying assumptions, hypothesis, or approximations (Desmarais, Maluf, Liu 1996). This also includes mathematically founded Models such as Bayesian modeling. The more assumptions and approximations are used, the less accurate and reliable a model becomes which in turn highly affects the reliability and sensitivity of the model. A consequent can be applicability of a model in only one context and poor outcome in another due to the violated assumptions.

1.4 Model of choice

The above-mentioned factors will determine the value and applicability of a student modeling approach. In an ideal case, a fully automated student model learning approach

should be able to use little data to build and calibrate and, yet, should provide detailed and accurate knowledge assessment. Such an approach would limit the effort of model building to that of data gathering and requires very few resources to update the model, such as adding new test items and new concepts. This ideal model, due to its algorithmic approach, is not dependent on environmental factors and therefore, has a considerable inherent reliability and accuracy.

Although it is obvious that the advantages of a learned model approach to student modeling are compelling, finding a universal model that meets all the above requirements is a hard to pin down goal. This can be better elaborated by looking at tradeoffs between some of these requirements. For example the larger the training data set is, the fewer assumptions need to be made and the better the model prediction accuracy will be. It is more appropriate to have a good understanding of circumstances under which a given model, for a specific task, is best suited. The above-mentioned dimensions could serve as a basis for investigating this issue.

The rest of this thesis is dedicated to investigation and comparison of two of these models, the POKS model based on item-to-item structures that can be learned from small data samples and Bayesian modeling with joint conditional probabilities. In this research we look for the tradeoffs between model parsimony and predictive accuracy. To accomplish that we compare a Bayesian Network (BN) model with a simple model based on the application of Bayes rule under strong independence assumption (POKS). The domain modeled is the mastery of individual skills.

Bayesian networks (BN), having the ability to represent complex relationships among variables with efficiency and parsimony, are considered highly powerful modeling and inferencing techniques. Bayesian modeling with joint conditional probabilities is conceptually and computationally the most straightforward mean of computing posterior probabilities.

However, in spite of these qualities, BN may not be always the most advantageous technique in comparison to simpler techniques that make stronger assumptions. For the same reason that BN offers parsimonious models for Bayesian modeling, thereby significantly increasing their usefulness in a wider range of application contexts, so do Bayes models with stronger independence assumptions. They offer more parsimonious representations for Bayesian modeling than do BN. However, they impose further assumptions on the domain model that can lead to invalid inferences.

POKS offers a simple framework for detailed knowledge assessment by estimating the mastery of individual knowledge items: an attempt towards learned student models. It offers fine-grained assessment at the item level, offering a learned statistical model, by making strong independence assumptions to meet the practical goals of full automation and the use of small data sets.

Our investigation will compare BN and POKS in the context of CAT. We will study how each frameworks performs for choosing the most informative items and for inferring skills based on previous answers. The difference between the context of CAT and a more general ILE context is that the system gets to choose items in CAT, whereas in the ILE context it may not have control of the evidence provided to infer skills. However, the goal remains the same: build a student model of skill mastery from evidence of other skills.

CHAPTER 2. PROBABILITY THEORY AND BAYESIAN NETWORK BASICS

Both POKS and the BN approach are grounded in Bayesian Theory. The following section is dedicated to explaining the basics of probability theory as well as Bayes' Theorem. We also move beyond the theory by explaining and providing simple example of Bayesian inference, which serves as a background for the two approaches.

2.1 The purpose of using probability theory

Bayesian networks (BN) are probabilistic graphical models that are rooted in acyclic graphs. They provide a tool to deal with two problems: uncertainty and complexity. Hence, they provide a compact representation of joint probability distributions using a combination of graph theory and probability theory. The graph structure specifies statistical dependencies among the variables and the local probabilistic models specify how these variables are combined.

Let us consider the situation in which one feature of an entity has a direct influence on another feature of that entity. For example, the presence or absence of a disease in a human being has a direct influence on whether a test for that disease turns out positive or negative. For decades, Bayes Theorem has been used to perform probabilistic inference in this type of situation. In the current example we would use that theorem to compute the ***conditional probability*** of an individual having a disease when a test for a disease came back positive.

Consider next the situation in which several features are related through inference chains. For example whether or not having a history of smoking in an individual has a direct influence both on whether or not that individual has bronchitis and on whether or not that individual has lung cancer. In turn the presence or absence of each of these diseases has a

direct influence on whether or not the individual experiences fatigue. Also, the presence or absence of lung cancer has a direct influence on whether or not a chest X-Ray is positive. In this situation, we would want to do ***probabilistic inference*** involving features that are not related via a direct influence.

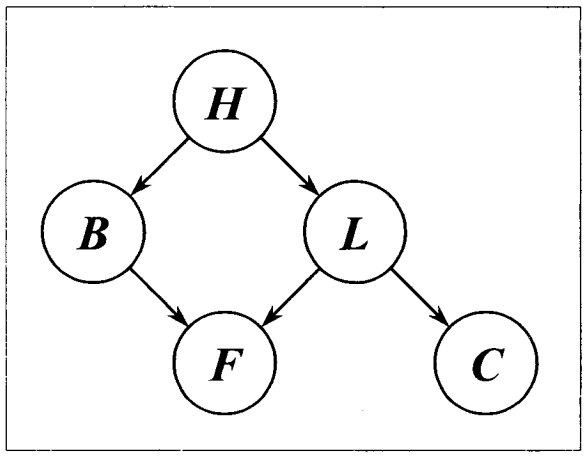


Figure 2.1. A simple Bayesian Network.

Table 2.1. Descriptions of the variables and their possible states in Lung Cancer example.

Variables	**Value**	**When the variable takes its value**
H	H1	There is a history of smoking
	H2	There is no history of smoking
B	B1	Bronchitis is present
	B2	Bronchitis is not present
L	L1	Lung cancer is present
	L2	Lung cancer is not present
F	F1	Fatigue is present
	F2	Fatigue is not present
C	C1	Chest X-ray is positive
	C2	Chest X-ray is negative

We would want to determine, for example, the conditional probabilities both of bronchitis and of lung cancer when it is known an individual smokes, is fatigued, and has a positive

chest X-Ray. Yet Bronchitis has no direct influence (indeed no influence at all) on whether a chest X-Ray is positive.

Bayesian networks were developed to address these inferences. By exploiting ***conditional independencies*** entailed by influence chains, a Bayesian Network enables us to represent large instances. In addition, the graphical nature of Bayesian networks gives us a much better intuitive grasp of the relationships among the features.

As briefly mentioned in the previous chapter, in addition to Bayesian networks there are other probabilistic approaches such as Partially Order Knowledge Structures (POKS) that can be conveniently represented graphically by a DAG that resembles to a Bayesian network. The semantics of links in POKS are different. Knowledge spaces links represent prerequisite, or, surmise relations (Doignon and Falmagne 1990). We return to Bayesian networks and POKS in more details later.

The following section explains the basics of the probability theory behind both methods, POKS and Bayesian Network. We elaborate the definition of ***probability function***, ***principle of indifference***, ***conditional probability***, as well as relationship between events such as ***Independent***, ***conditionally independent***, ***mutually exclusive and exhaustive events***. These definitions and the underlying mathematics are the base behind the inferences used by the above-mentioned methods.

2.2 Mathematical foundation of probability theory

In 1933, Kolmogorov developed the set-theoretic definition of probability, which serves as a mathematical foundation for all applications of probability. Probability theory has to do with experiments that have a set of distinct ***outcomes***. For the sake of our discussion here, these outcomes need to be ***distinct*** and ***infinite***. Once an experiment is well defined, the collection of outcomes is called the ***sample space***. Suppose we have a sample space containing n elements. That is

$$\Omega = \{e_1, e_2, ..., e_n\} \tag{2.1}$$

A function that assigns a real number $P(E)$ to each event $E \subset \Omega$ is called a ***probability function*** on the set of subsets of Ω satisfying:

1. $0 \le P(\{e_i\}) \le 1$ for $1 \le i \le n$
2. $P(\{e_1\}) + P(\{e_2\}) + ... + P(\{e_n\}) = 1$
3. For each event $E = \{e_{i1}, e_{i2}, ..., e_{ik}\}$ that is not an elementary event, $P(E) = P(\{e_{i1}\}) + P(\{e_{i2}\}) + ... + P(\{e_{ik}\})$ (2.2)

The Pair (Ω, P) is called ***probability space***.

To assess the probability of items we need to refer to the ***principle of indifference*** (J.M. Keynes 1921), which says elementary events are to be considered equi-probable, if we have no reason to expect to prefer one to the other. Therefore when there are n elementary events, the probability of each item is the ***ratio*** $1/n$.

A useful theorem concerning probability space is called ***the axioms of probability theory*** (Kolmogorov 1933) as:

1. $P(\Omega) = 1$
2. $0 \le P(E) \le 1$ for every $E \subset \Omega$
3. For E and $F \subset \Omega$ such that $E \cap F = \phi$, $P(E \cup F) = P(E) + P(F)$ (generally shown as: $P(E \cup F) = P(E) + P(F) - P(E \cap F)$) (2.3)

2.3 Conditional Probability

Another notable concept in probability is called ***conditional probability***: Let E and F be events such that $P(F) \neq 0$. Then the conditional probability of E given F, $P(E|F)$, is given by

$$P(E|F) = \frac{P(E \cap F)}{P(F)} \tag{2.4}$$

The initial intuition for conditional probability comes from considering probabilities that are ratios. In the case of ratio, $P(E|F)$, as defined above, is the fraction of items in F that are also in E. Let n be the number of items in the sample space, n_f be the number of items in F, and n_{ef} be the number of items in $E \cap F$. Then

$$\frac{P(E \cap F)}{P(F)} = \frac{n_{ef}/n}{n_f/n} = \frac{n_{ef}}{n_f} \tag{2.5}$$

which is the fraction of items in F that are also in E. *As far as meaning, P (E | F) means the probability of E occurring given that we know F has occurred.*

Two other important aspect in probability domain and Bayesian networks are ***Independent*** and ***conditionally independent events***, which are defined as below:

Two events E and F are ***independent*** if one of the following holds:

1. $P(E \mid F) = P(E)$ and $P(E) \neq 0, P(F) \neq 0$
2. $P(E) = 0$ or $P(F) = 0$ (2.6)

An important consequence can be that E and F are independent if and only if:

$$P(E \cap F) = P(E) * P(F) \tag{2.7}$$

Two events that are ***conditionally independent*** can be defined as:

$$\begin{aligned} &1.\quad P(E \mid F \cap G) = P(E \mid G) \quad \text{and} \quad P(E \mid G) \neq 0, P(F \mid G) \neq 0 \\ &2.\quad P(E \mid G) = 0 \quad \text{or} \quad P(F \mid G) = 0 \end{aligned} \tag{2.8}$$

A very useful rule involving conditional probability says that suppose we have n events E_1, E_2, to E_n such that $E_i \cap E_j = \phi$ for $i \neq j$ and $E_1 \cup E_2 \cup E_3 \cup ... \cup E_n = \Omega$. Such events are called ***mutually exclusive and exhaustive***. Then the law of total probability says that for any event other than F,

$$P(F) = \sum_{i=1}^{n} P(F \cap E_i). \tag{2.9}$$

If $P(E_i) \neq 0$ then $P(F \cap E_i) = P(F|E_i)P(E_i)$. Therefore, if $P(E_i) \neq 0$ for all i, the law is often applied in the following form:

$$P(F) = \sum_{i=1}^{n} P(F|E_i)P(E_i). \tag{2.10}$$

These are the basic probability definitions behind the Bayesian framework. The following section describes the Bayes Theorem and how the above-mentioned definitions are applied to Bayesian networks.

2.4 Bayes theorem

For decades conditional probabilities of events of interest have been computed from known probabilities using Bayes' theorem:

Given two events E and F such that $P(E) \neq 0$ and $P(F) \neq 0$, we have:

$$P(E|F) = \frac{P(F|E)P(E)}{P(F)} \tag{2.11}$$

Furthermore, given n mutually exclusive and exhaustive events E_1, E_2, ...E_n such that $P(E_i) \neq 0$ for all i, we have for $1 \leq i \leq n$,

$$P(E_i|F) = \frac{P(F|E_i)P(E_i)}{P(F|E_1)P(E_1) + P(F|E_2)P(E_2) + ... + P(F|E_n)P(E_n)}. \qquad (2.12)$$

Both of the formulas above are called ***Bayes theorem*** that are developed originally by Thomas Bayes (1763).

2.5 Random Variables and Joint Probability Distribution

Furthermore we need to have a clear understanding of ***Random Variable*** and ***Joint Probability Distribution***. Given a probability space (Ω, P), a *random variable* X is a function on Ω. That is a random variable assigns a unique value to each element (outcome) in the sample space. The set of values that random variables X can assume is called the ***space*** of X. A random variable is set to be ***discrete*** if its space is finite or countable (which is what we assume in general).

Here we explain this notion in an example: If Ω contains all outcomes of a throw of a pair of six-sided dice, and let P assign 1/36 to each outcome then:

$$\Omega = \{(1,1), (1,2)\ (1,3)\ (1,4)\ (1,5)\ (1,6)\ (2,1)\ (2,2)\ (2,3), \ldots, (6,5)\ (6,6)\}. \qquad (2.13)$$

Let the random variable X be the "sum" of each ordered pair to that pair, and let the random variable Y be "odd" for each pair of odd numbers and "even" for a pair if at least one number in that pair is an even number. The Following table shows some of the values of X and Y:

Table 2.2.Values for X (sum) and Y (odd, even) functions

E	X(e)	Y(e)
(1,1)	2	odd
(1,2)	3	even
...	...	...
(2,1)	3	even
...	...	...
(6,6)	12	even

Therefore: The space of X is {2,3,4,5,6,7,8,9,10,11,12} and that of Y is {odd, even}.

In another example let Ω contain all the outcomes of a throw of a single die, Let P assign 1/6 to each outcome and let Z assign "even" to each even number and "odd" to each odd number. Then

$$P_z(\{even\}) = P(Z = even) = P(\{2,4,6\}) = 1/2 \tag{2.14}$$

$$P_z(\{odd\}) = P(Z = odd) = P(\{1,3,5\}) = 1/2 \tag{2.15}$$

Here P is the original probability function or ***probability distribution*** in this case. We refer to $P(X)$ (instead of $P(X = x)$) as ***"the probability of x"***. Therefore for the cases of $x = 3$ in our first example, pairs of (1,2) and (2,1), we have $P(x) = P(X = x) = 1/18$.

Another important concept in probability theory is the ***expected value***. If X is a discrete random variable with values x_1, x_2, ... and corresponding probabilities p_1, p_2, ... which add up to 1, then expected value, $E(X)$, can be computed as the sum or series:

$$E(X) = \sum_i p_i x_i \tag{2.16}$$

For example let Ω contain all outcomes of a throw of a single die, let P probability of each come is 1/6 and let x is the value of the outcome (number appearing on the die). Then:

$$E(X) = \sum_{x=1}^{6} xP(X) = \sum_{x=1}^{6} x(\frac{1}{6}) = \frac{7}{2}. \qquad (2.17)$$

Given two random variables X and Y, defined on the same sample space Ω, we use $X = x$ and $Y = y$ to denote the set of all elements $e \in \Omega$ that are mapped both by X to x and by Y to y. Referring to our example, $X = 4$ and $Y =$ odd represents the event {(1, 3), (3, 1)} and $P(X = 4, Y = odd) = 18$.

Clearly two random variables induce a probability function on the Cartesian product of their space. We refer to $P(X = x, Y = y)$, and we call this the ***joint probability distribution*** of X and Y.

Given a joint probability distribution, the law of ***total probability*** implies that the probability distribution of any one of the random variables can be obtained by summing over all values of the other variables. For example, suppose we have a joint probability distribution $P(X = x, Y = y)$ then:

$$P(X = x) = \sum_{y} P(X = x, Y = y). \qquad (2.18)$$

The probability distribution $P(X = x)$ is called the ***marginal probability distribution*** of X because it is obtained using a process similar to adding across the row or column in a table of numbers. This concept also extends in a straightforward way to three or more random variables.

Suppose we have a probability space (Ω, P) and two sets A and B random variables defined in Ω. Then the sets A and B are said to be ***independent*** if, for all values of the

variables in the sets, a and b the events $A = a$ and $B = b$ are independent. That is either $P(a) = 0$ or $P(b) = 0$ or $P(a|b) = P(a)$. This can be written as $I_P(A, B)$ where I_P stands for Independent in P.

Events can also be ***conditionally independent***: Suppose that we have a probability space (Ω, P), and three sets A, B and C containing random variables defined on Ω. The sets A and B are said to be *conditionally independent* given that the set C if, for all values of the variables in the sets a, b and c, whenever $P(c) \neq 0$ the events $A = a$ and $B = b$ are *conditionally independent* given the event $C = c$. That is, either $P(a|c) = 0$ or $P(b|c) = 0$ or $P(a|b, c) = P(a|c)$

2.6 Relative Frequency (Von Mises theory) vs. Subjective (Bayesian)

Statistical regularity has motivated the development of the ***relative frequency*** concept of probability. Most of the procedures commonly used to make statistical estimates or tests were developed by statisticians who used this concept exclusively. A statistician who uses traditional methods of inference is therefore possibly referred to as a ***Frequentist*** statistician (Von Mises 1928). Frequentists insisted that statistical procedures only made sense when one uses the relative frequency concept; on the contrary, the ***Bayesians*** supported the use of degrees of belief as a basis for statistical practice.

For example based on the Frequentist position you can perform an experiment lots of times, and measure the proportion where you get a specific result. This proportion, if you perform the experiment enough times, is the probability.

The problem comes from those cases where we have yet to perform an experiment, or where there is no possible way an experiment could be performed - in these cases, frequentism is of no help.

In his book Theory of Probability Bruno de Finetti (De Finetti 1926) argued that probabilities should be treated as appropriately calibrated degrees of belief, with a one-to-one relationship to betting odds. Using betting as an analogy for all decision-making under uncertainty, he proved that necessary and sufficient conditions for rational betting (i.e. all decision-making under uncertainty) were that subjective degrees of belief which satisfies the Kolmogorov axioms for probability. This laid the foundation of an alternative version of probability, now known as Bayesian.

Therefore for the cases such as ascertaining the probability of winning of Montreal Canadians in a hockey game and similar cases, the probability is not a ratio and is not a relative frequency or even an estimate of relative frequency (because you can not repeat the game many times) rather the probability represents the belief concerning the chance of having certain outcomes.

A probability such as this is called a ***degree of belief*** or ***subjective probability***.

The subjective probability approach is called "***Bayesian***" because its proponents use Bayes' theorem to infer unknown probability from known ones. There are two concepts to clarify before using the theorem: ***Prior probability*** and ***posterior probability***.

A probability is called a *prior probability* because, in a particular model, it is the probability of some event prior to updating the probability of that event, within the framework of that model, using new information. A *prior probability* is a marginal probability, interpreted as a description of what is known about a variable in the absence of some evidence.

A probability is called a *posterior probability* because it is the probability of an event after its prior probability has been updated, within the framework of some model, based on new information. In Bayesian probability theory, the *posterior probability* is the conditional probability of some event or proposition, taking empirical data into account

One applies Bayes' theorem, multiplying the prior by the ***Likelihood function*** and then normalizing, to get the ***posterior probability distribution***, which is the conditional distribution of the uncertain quantity given the data.

2.7 Bayesian interpretations

Good (1983) shows there are 46656 different Bayesian interpretations. Briefly, there is a ***descriptive Bayesian interpretation*** which maintains that human reason using subjective probabilities and Bayes theorem, there is a ***normative Bayesian interpretation*** that says human should reason that way, and there is an ***empirical Bayesian interpretation*** that says, based on data, we can update our belief concerning a relative frequency using Bayesian theorem. The rest of this thesis uses the later interpretation.

2.8 Large instances /Bayesian networks

Bayesian inference is fairly simple when it involves only two related variables. However it becomes much more complex and requires numerous calculations when we want to do inference with many related variables. This can be solved by describing the Markov condition (explained in the following section), which is a relationship between graphs and probability distributions. Bayesian network exploits the Markov condition in order to represent large instances efficiently. Bayesian networks address these problems representing the joint probability distribution of a large number of random variables and with implementing Bayesian inference with these variables.

2.9 The Markov conditions

We know that a ***directed graph*** is a pair of (V,E), where V is finite, nonempty set whose elements are called ***nodes*** or ***vertices***, and E is a set of ordered pairs of distinct elements of V. Elements of E are called ***edges*** or ***arcs***. A ***directed cycle*** is path from node to itself. A directed graph G is called a ***directed acyclic graph (DAG)*** if it contains no directed cycles. Given a DAG $G=(V,E)$ and nodes X and Y in V, Y is called a ***parent*** of X if

there is an edge from Y to X, Y is called a ***descendant*** of X and X is called an ***ancestor*** of Y if there is a path from X to Y, and Y is called a ***no descendant*** of X if Y is not a descendant of X. There is a very important definition, which is called ***Markov Condition*** and is used in construction of a large Bayesian network:

Suppose we have a joint probability distribution P of the random variables in some set V and a DAG G=(V,E). We say that (G,P) satisfies the Markov Condition if for each variable $X \in V$, {X} is conditionally independent of the set of all its non-descendants given the set of all it parents.

We know that the number of terms in a joint probability distribution is exponential in terms of the number of variables. So in the case of a large instance, we could not fully describe the joint distribution by determining each of its values directly. Herein lies one of the powers of Markov condition, it is possible to prove that:

If (G,P) satisfies Markov Condition, then P equals the product of its conditional probability distributions of all nodes given values of their parents in G, whenever these conditional distribution exist. *(Theorem 2.1)*

This means that we often need to ascertain far fewer values than if we had to determine all values in the joint distribution directly. This will be the case for a joint probability distribution P of the variables in the DAG presented in Figure 2.2 if, for all values of f, c, b, l and h, $P(f,c,b,l,h) = P(f|b,l)P(c|l)P(b|h)P(l|h)P(h)$, whenever the conditional probabilities on the right exist.

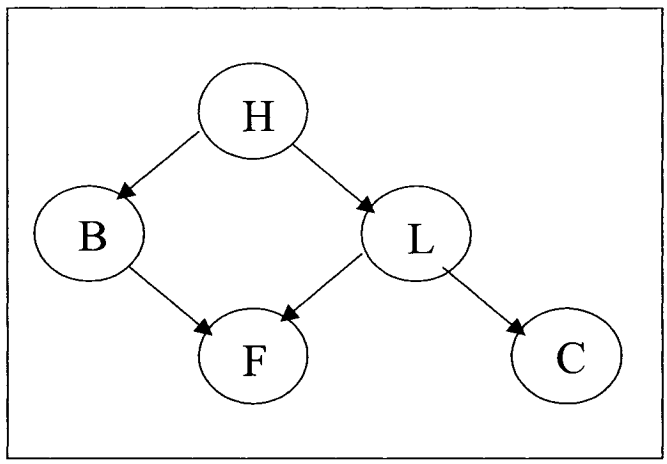


Figure 2.2. An example of DAG created for joint probability distribution.

This theorem enables us to reduce the problem of determining a huge number of probability values to that of determining relatively few. The number of values in the joint distribution is exponential in terms of the number of variables. However due to this theorem each of these values is uniquely determined by the conditional distributions and if each node in the DAG does not have too many children, there are not many values in these distribution. For example, if each variable has two possible values and each node has at most one parent we would need to ascertain less than $2 \times n$ probability values to determine the conditional distribution when the DAG contains n node. On the other hand we would need to ascertain $2^n - 1$ values to determine the joint probability distribution directly. In general, if each variable has two possible values and each node has at most k parents, we need to ascertain $2^k \times n$ values to determine the conditional distributions. So if k is not large, we have a manageable number of values.

This approach will be good if we start with an underlying sample space and probability function, specify some random variables and show that if P is the probability distribution of these variables and G is the DAG then (P,G) satisfies the Markov condition. Then by applying the theorem (1) we can conclude that we need to only determine the conditional distributions of the variables for that DAG and to find any values in the joint distribution. However in real application we do not ordinarily specify an underlying sample space and

probability function from which we can compute conditional distribution. Rather we identify random variables and values in conditional distributions directly. For example, in an application involving the diagnosis of lung cancer, we identify variables like *smoking history*, *lung cancer*, and *chest X ray*, and probabilities such as P(*SmokingHistory* = *yes*), P(*LungCancer* = *present* | *SmokingHistory* = *yes*), and P(*ChestXRay* = *positive* | *LungCancer* = *present*). Here we don't know if the product of these conditional distributions is a joint distribution at all, neither if it satisfies the Markov condition. Therefore we need to work in reverse. We must start with the conditional distribution and then be able to conclude the product of these distributions is a joint distribution satisfying the Markov condition with some DAG.

To do that we can use the following theorem:

Let a DAG G be given in which each node is a random variable, and let a discrete conditional probability distribution of each node given values of its parents in G be specified. Then the product of these conditional distributions yields a joint probability distribution P of the variables, and (G,P) satisfies the Markov conditions with some DAG. *(Theorem 2.2)*

2.10 Bayesian networks

Now that the entire requirements for construction of a BN is ready, we can define one:

Let P be a joint probability distribution of the random variables in some set V, and G = (V,E) be DAG. We call (G,P) a Bayesian network if (G,P) satisfies the Markov Condition.

Based on the Theorem (2.1), *P* is the product of its conditional distributions in *G* and this is the way that p is always represented in a Bayesian network. Furthermore owing to the Theorem (2.2), if we specify a DAG G and any discrete conditional distribution (and

many continuous ones), we obtain a Bayesian network. This is the way Bayesian networks are constructed in practice.

There are two components of a BN model: $M = \{G, P\}$. Each node in the graph G represents a random variable and edges represent conditional independence relationships. The set P of parameters specifies the probability distributions associated with each variable. Edges represent "causation" so no directed cycles are allowed. Markov property is also obeyed: Each node is conditionally independent of its ancestors given its parents.

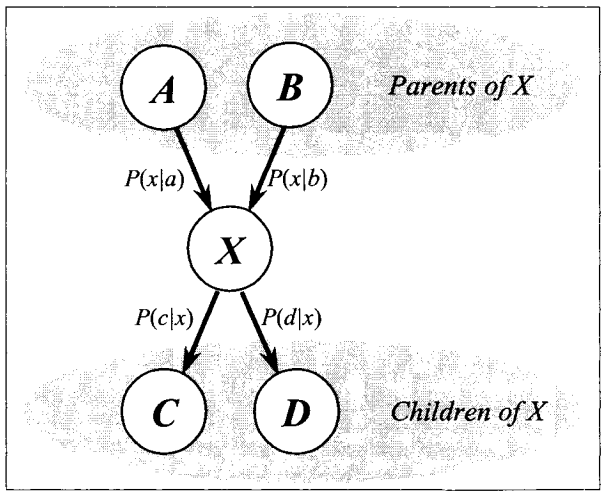


Figure 2.3. Simple Probability inference in a Bayesian network

The joint probability of a set of variables $x_1,...,x_n$, using the chain rule can be given as:

$$P(X_1,....,X_n) = \prod_{i=1}^{n} P(x_i|x_1,....,x_{i-1}) \tag{2.19}$$

The conditional independence relationships encoded in the Bayesian network state that a node x_i is conditionally independent of its ancestors given its parents π_i. Therefore,

$$P(X_1,....,X_n) = \prod_{i=1}^{n} P(x_i|\pi_i) \tag{2.20}$$

Once we know the joint probability distribution encoded in the network, we can answer all possible inference questions about the variables using marginalization. The following section provides different examples of the Bayesian networks.

2.11 Bayesian Network example

The following figures provide examples of Bayesian network including their inferences.

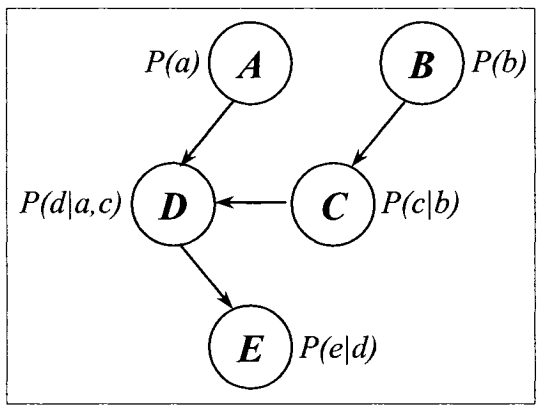


Figure 2.4. $P(a,b,c,d,e) = P(a)P(b)P(c|b)P(d|a,c)P(e|d)$.

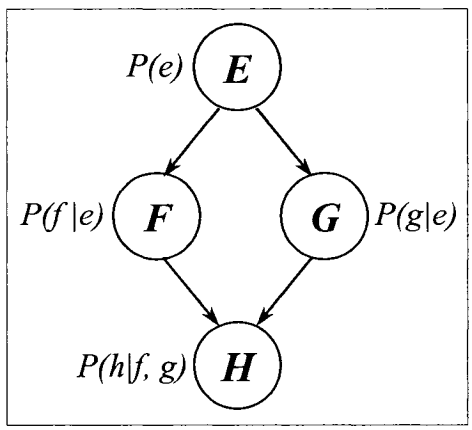


Figure 2.5. $P(e,f,g,h) = P(e)P(f|e)P(g|e)P(h|f,g)$.

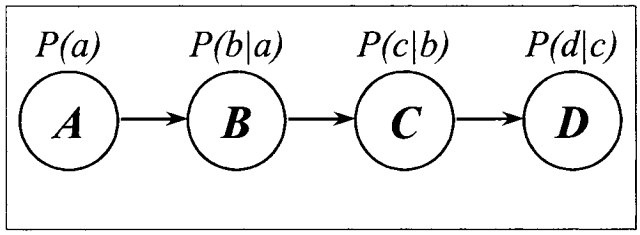


Figure 2.6. $P(a,b,c,d) = P(a)P(b|a)P(c|b)P(d|c)$

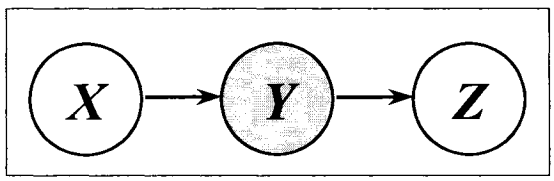


Figure 2.7. Y is given, X and Z are conditionally independent. X=past, Y =present and Z=future.

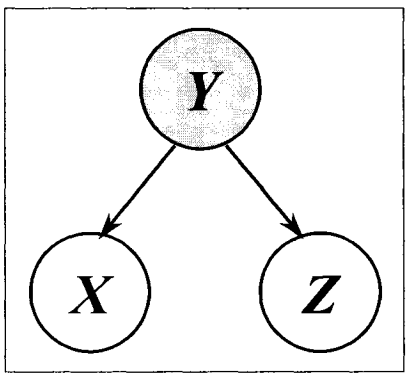


Figure 2.8. When Y is given, X and Z are conditionally independent. Y is the common cause of the two independent effects X and Z.

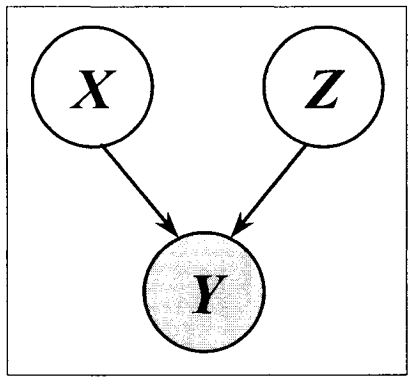


Figure 2.9. X and Z are marginally independent, but when Y is given, they are conditionally dependent.

Here we provide a concrete example of Bayesian networks. Lets assume that you have a new burglar alarm installed at home and:

- It is fairly reliable at detecting burglary, but also sometimes responds to minor earthquakes.
- You have two neighbors, Alex and Tamer, who promised to call you at work when they hear the alarm.
- Alex always calls when he hears the alarm, but sometimes confuses telephone ringing with the alarm and calls too.
- Tamer likes loud music and sometimes misses the alarm.
- Given the evidence of who has or has not called, we would like to estimate the probability of a burglary.

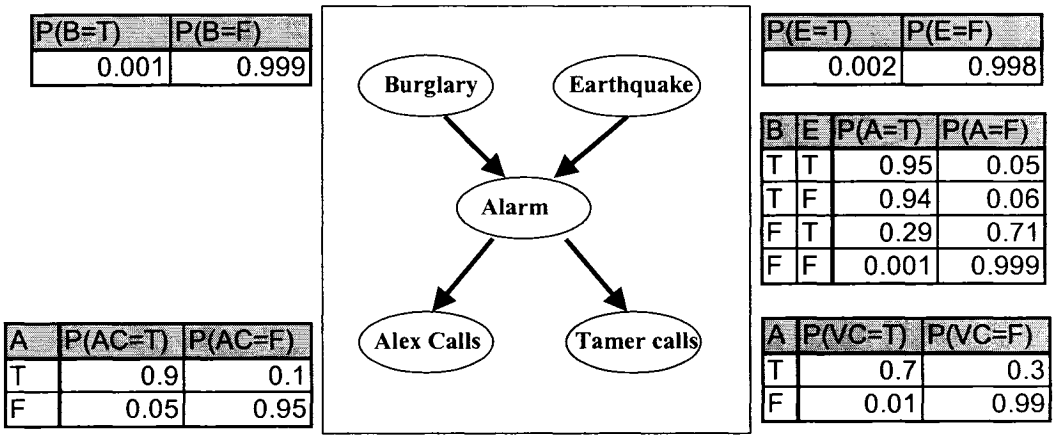


Figure 2.10. The Bayesian network for the burglar alarm example.

Burglary (B) and earthquakes (E) directly affect the probability of the alarm (A) going off, but whether or not Alex calls (AC) or Tamer calls (VC) depend only on the alarm. Now that we have developed the structure for a Bayesian network, we start asking question about the network: What is the probability that the alarm has sounded but neither a burglary nor an earthquake has occurred, and both Alex and Tamer call?

$$
\begin{aligned}
&P(AC, VC, A, \sim B, \sim E) \\
&= P(AC|A) \times P(VC|A) \times P(A|\sim B, \sim E) \times P(\sim B) \times P(\sim E) \\
&= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998 \\
&= 0.00062
\end{aligned}
$$

(Capital letters represent variables having the value true, and ~ represents negation)

What is the probability that there is a burglary given that Alex calls?

$$P(B|AC) = \frac{P(B, AC)}{P(AC)}$$

$$= \frac{\sum_{VC}\sum_{A}\sum_{E} P(AC|A)P(VC|A)P(A|B,E)P(B)P(E)}{P(B, AC) + P(\sim B, AC)}$$

$$= \frac{0.00084632}{0.00084632 + 0.0513}$$

$$= 0.0162$$

What about if Tamer also calls right after Alex hangs up?

$$P(B \mid AC, VC) = \frac{P(B, AC, VC)}{P(AC, VC)} = 0.29$$

2.12 Conclusion

In this chapter we explored the mathematical foundation behind the probability as well as Bayesian decision theory. By Investigating its root in statistics and graph theory, we recognized that Bayes theorem provides methodology for rational choice under uncertainty and combines prior knowledge with observed data. We also briefly discussed Bayesian Network They are generally more compact than the full joint probability distribution: For *n* Boolean random variables and having the maximum number of parents of a node in a network as *k*, Bayesian net compares less (size) complex with the full joint probability distribution: Bayesian net with complexity of $n \times 2^k$ compare to Joint probability distribution as 2^n.

Approaches such as Bayesian networks (BN) are considered highly powerful modeling and inferencing techniques because they make few assumptions and they can represent complex relationships among variables with efficiency and parsimony. They can also be learned from training data, and generally lend themselves to a variety of sound and efficient inference computations.

However, in spite of these qualities, BN may not be always the most advantageous technique in comparison to simpler techniques that make stronger assumptions. Although probabilistic graph models provide a means to perform detailed knowledge assessment,

but this comes at a price: their drawback is that they often require a large effort to build. Furthermore, the success of that effort depends upon the availability of an often large dataset, on the ability of an expert who must understand the knowledge domain and the probabilistic techniques involved.

In the following two chapters we investigate these issues in the domain of skills modeling and assessment. We will investigate two models: First a simple Bayes posterior probability update approach under strong independence assumptions, named POKS (Desmarais, Maluf, and Liu 1995) and then Vomlel's (2004) BN model of basic arithmetic skills. We compare these two models that are rooted in the probabilistic theory, on the basis of predictive accuracy and evaluate the advantages of each and combined.

CHAPTER 3. POKS PARTIAL ORDER KNOWLEDGE STRUCTURE

This chapter discusses the basics behind Partial Order Knowledge Structure, POKS, as a method to assess the mastery and non-mastery of a subject in educational assessment. The method explained here will be utilized later in simulations and compared with standard Bayesian networks in the domain of knowledge assessment.

3.1 User expertise modeling tool POKS

Most intelligent student models are organized in a hierarchy of concepts with observable nodes, mainly test items, as leaves of this hierarchy. The "non observable" nodes are concepts, skills, and misconceptions. The relations are not always in hierarchy and can also include relations linking sibling nodes, or children nodes to multiple parents. Yet, the general underlying structure remains hierarchical.

Some approaches link observable nodes among themselves to build structures that do not contain any hidden nodes (concept nodes). By doing so, they are meant to be models for predicting item responses outcome, not direct models of skills. However, it is a routine task for any teacher to derive skills from responses to an exam.

Examples of approaches that rely on item to item structures are (Dowling and Hockemeyer 2001, Kambouri, Koppen, Villano, and Falmagne 1994, Desmarais *et al.* 1996). As explained before, they come from the work of Falmagne, Koppen, Villano, Doignon and Johannesen (1990) and Doignon and Falmagne (1999). This is the formalism for the representation of the order in which we learn knowledge units (KU) and the theory of knowledge spaces. The work on Partial Order Knowledge Structures (POKS) (Desmarais *et al.* 1996, Desmarais and Pu 2005) falls under this line of research as well.

POKS allows the induction of knowledge structures from a small number of empirical data cases. It uses an evidence propagation scheme within these structures to infer an individual's knowledge state from a sample of KU. The empirical induction technique is based on statistical hypothesis testing over conditional probabilities that are determined by the KUs' learning order. Previous experiments with this approach has shown that the technique is successful in partially inferring an individual's knowledge state, either through the monitoring of a user's behavior, or through a selective questioning process.

3.2 Major characteristics of POKS

In general we can summarize major characteristics of POKS as following:

- POKS permits the inference of the structure among item nodes.
- POKS defines possible knowledge states closed under union and intersection, whereas AND/OR graphs define possible knowledge states closed under union only.
- POKS makes the assumption of local independence among evidence nodes
- POKS essentially makes the assumption that we can limit the modeling solely to binary conditional probability relations (It allows the induction of the network from a very small number of data cases).
- The network of POKS is defined solely over the test items and no concepts nodes are included. Imposing this rule relieves POKS from any knowledge engineering effort to construct the network.
- The evidence propagation scheme is unidirectional in the sense that if a node's probability increases, no backward propagation is performed, and, conversely, no forward propagation is performed when a node's probability decreases.

3.3 Knowledge structures in POKS

Based on the theory of knowledge spaces knowledge items, i.e. observable elements that define a knowledge state such as question items, are mastered in a constrained order. This defines the structure of prerequisites among knowledge items. As an example, in order to solve problem presented in figure 3.1, we use an order that complies with the *inverse* of the arrow directions. Basically if one succeeds knowledge item (c), it is likely she will also succeed item (d). On the other hand, if she fails item (c), she will likely fail item (a). However, item (c) provides no significant information about item (b). This structure defines the following possible knowledge states (subsets of the set $\{a,b,c,d\}$): $\{\emptyset,\{d\},\{d,c\},\{d,b\},\{a,b,c,d\},\{d,b,c\}\}$

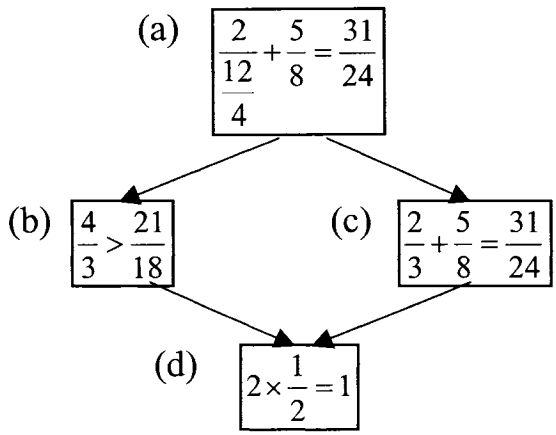


Figure 3.1. A simple knowledge space composed of 4 items ($\{a,b,c,d\}$) with a partial order that constrains possible knowledge states to $\{\emptyset,\{d\},\{d,c\},\{d,b\},\{d,b,c\},\{a,b,c,d\}\}$

A significant contribution to the formal representation of the interdependencies among KU was the work of Falmagne *et al.* (1990) on knowledge structures and knowledge spaces. In their work, they have shown that the constraints on the order in which we learn KU can be entirely represented by AND/OR graphs.

3.3.1 Inference rules and AND/OR graph

If we want to specify AND/OR graphs in terms of inference rules, we need to use these following two rules:

1. "If A is known, then B, and C, ... and N are known",
2. "If A is known, then either B, or C, ... or N, is known",

where each rule has only one antecedent and one or more consequents. Each consequent has an arc from the antecedent to the consequent, which shows the ordering constraint. The second rule, is obviously the one that can be achieved via the 'OR' operator in the 'AND/OR' graph representation. It is used where a given knowledge states can be reached through a number of different knowledge states. For example, creating a loop in the "C" programming language requires mastery of the "for", "while", or "do" constructs, but only one of them is required to reach the knowledge state, the ability to create loops.

Therefore, for every two potential knowledge states reached by a user, the union of these two states is also a potential knowledge state. This condition corresponds to the definition of a closure in the space of knowledge states under the union operator, $\bigcup$.

In other words, if we combine two individuals' knowledge states, then that combined knowledge state is also plausible. However, knowledge spaces are not closed under intersection, meaning that if we take the common knowledge items between two individuals' knowledge states, then we can obtain an invalid knowledge state. This phenomenon occurs when a knowledge item has two alternative prerequisites. For example, one individual might learn to add two fractions by first transforming them into a common denominator, whereas someone else might have learned to transform them into decimal form first and back into a rational form. If each of them ignores the other individual's method, then the intersection of their knowledge states yields a state with the mastery of the fraction addition problem with none of the other two prerequisite knowledge items are mastered.

In the case of POKS, we make the assumption/approximation that knowledge spaces are closed under union and intersection and ignore the possibility of representing alternate prerequisite knowledge items. We refer to this variant as partial order knowledge structures, or POKS. Such structures can be represented by a DAG, such as the one in figure 3.1 because we further impose the assumption of closure under intersection (Desmarais *et al.* 1996). This assumption has strong implications on the reduction of the data set size required.

Looking at the above mentioned example (loop in C^{++}), we observe that it is possible to have a programmer that succeeds in an exercise involving loops but knows only the "for" construct, and someone else who also succeeds at the same exercise but only knows the "while" loop. Therefore, closure under both $\bigcup$ and $\bigcap$ could be violated. The intersection of these two knowledge states represents a subject who can solve a problem involving loops without knowledge of any of the corresponding syntactic constructs. It is impossible to represent this type of disjunctive relationship among KU with partial orders where closure under $\bigcup$ and $\bigcap$ is assumed.

Although partial orders are simpler and less powerful formalisms than AND/OR graphs, they are nevertheless very useful and have played a much greater role in user modeling. They were used by a number of researchers to represent the implication relations among KU (Goldstein 1982, Burton 1982, Bretch and Jones 1988).

In POKS, the user's knowledge state is represented as a subset of a global set of KU. Moreover, the global set of KU is interconnected with a number of implication relations. Relations, such as $A \Rightarrow B$, allow inferences of the type.

1. "if A is known, then B must be known" and
2. "if B is unknown, then A is unknown".

If there is a strongly connected structure of KU, the process of assessing someone's knowledge state can be highly efficient, therefore a few KU need to be known in order to draw conclusions about the complete knowledge state.

The above mentioned implication structure among KU can be determined by the order in which we learn concepts. This forms one of the most important characteristics of the general learning process, a well-known phenomenon in education (Gagné 1966).

3.4 POKS in expertise modeling

The POKS approach has the formal properties of DAG. Although it does not have the ability to represent alternative means of reaching a given knowledge state, it has great significance in expertise modeling:

- Generally, a knowledge item has many more fixed prerequisites than alternative prerequisites, such that a POKS will contain most of the underlying structure in a knowledge domain.
- The relations in a POKS are transitive, a desirable feature for the knowledge assessment process and for building parsimonious knowledge structures.
- A POKS can also contain probabilistic information that will capture some of the information found in alternative prerequisites: alternative prerequisites will be represented in the structure as "weaker" (in a probabilistic sense) prerequisites than would fixed prerequisites.

Therefore we can conjecture that POKS is very useful in assessing someone's knowledge state, which is the goal of user modeling. An example to illustrate the notions behind a POKS is given in figure 3.2. It contains a graphical representation of a POKS for UNIX shell commands.

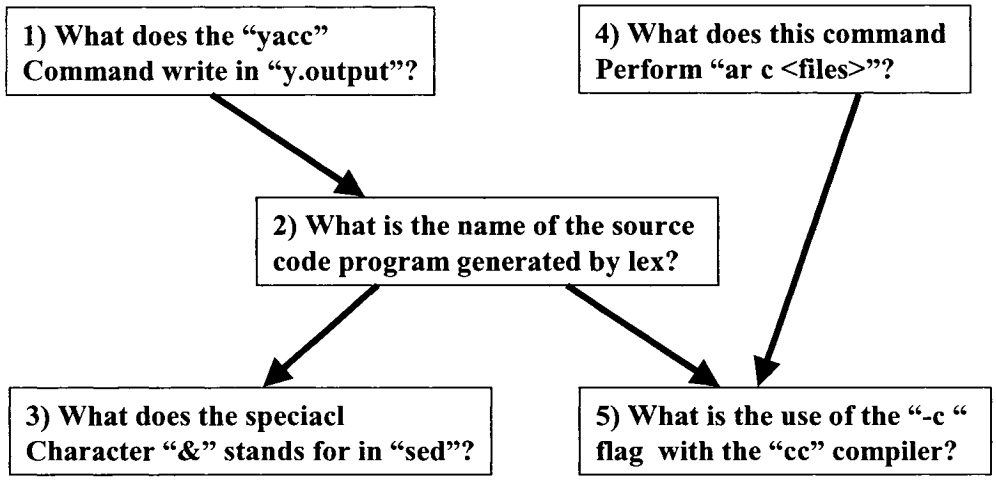


Figure 3.2. Inference network with Unix Command Knowledge Units (KU).

Nodes are numbered to be referenced later. First node asks question about "Yacc". "Yacc" is a parser program generator. It generates a parser in "C" source code and it is generally used in conjunction with "lex" which does the lexical analysis part. "y.output" is the file containing the parsing table information. The next question is about "lex". "lex" is the lexical analyzer. It also generates "C" source code and writes it in a file named "lex.y.c". The third question discusses the "sed" command. "sed" is a string editor for performing string manipulations on whole files. The '&' stands for the whole string matched. The fourth question, which somehow is independent from the others, talks about the "ar" command. "ar" creates library archives and the "ar c <files>" creates the archive file. Finally the "cc" command is the "C" source code compiler and the "c" flag is used for creating object files.

We use the arcs (as explained in previous chapters) between the nodes to represent surmise relations within the structure. Figure 3.2 shows that some of these relations contain strict prerequisites, such as (4) $\Rightarrow$ (5) and (1) $\Rightarrow$ (2). It is necessary to know about the "c", the compiler flag, in order to generate archive files. Generally speaking, programs generated with "yacc" will use "lex" as a lexical analyzer preprocessor.

But looking at the other surmise relations we can see that they are of an empirical nature, namely (2) ⇒ (5) and (2) ⇒ (3): knowledge of "sed" or of the 'C' compiler's "c" flag is not a prerequisite to using "lex", but it is generally the case that these two KU (3 and 5) will be learned before KU no. 2. The relations (1) ⇒ (5) and (1) ⇒ (3) are not explicitly specified because they can be derived from transitivity.

Regardless if the surmise relations are determined by prerequisites, or by other empirical factors that in a way puts a constrain on the order of learning among KU, it must be emphasized that the order may be violated in a number of cases, either because of noise in the assessment of mastery, or because the surmise relation is "weak" and there are many exceptions.

Nevertheless, the surmise relation should not be ignored because of the noise or the exceptions. *Instead, it should justify the use of a stochastic approach for modeling this phenomenon.*

There are number of potential architectures for building knowledge structures. Regardless of the chosen architecture, the important factors defining a relevant set of KU are:

- KU do represent meaningful and significant units in the domain of knowledge;
- The user's mastery of each KU can be reliably assessed, and
- There is some order in the way users learn KU.

The domain expert's ability to break down the knowledge domain into KU heavily affects the first two factors. As an example, we can name the ability that one requires for developing a good final exam. This relates to the theory of psychological testing, for which a large body of theory and practice already exists (Anastasi 1966). Ordering of KU varies across different knowledge domains, but it is a fairly ubiquitous learning phenomenon. *The less any of the three factors above is valid, the less the POKS will contain relations, and the less effective and reliable it will be in assessing a user's knowledge state.*

The above example is useful in explaining the theory of POKS and its application to user modeling. Let us now define more precisely the notions involved.

3.4.1 Definitions for POKS

Lets assume that we have a knowledge domain, denoted Q, composed of n KU:

$$Q = \{U_1, U_2, ..., U_n\} \tag{3.1}$$

We define a POKS as a Partial Order over Q where the nodes represent the domain's KU, $\{U_1, U_2, ..., U_n\}$, and the arcs represent implications, or surmise relations.

An arc from U_i to U_j, is denoted as $S_{i \Rightarrow j}$. R as shown below, denotes an individual's knowledge state. Based on standard overlay model representation, the knowledge state can be represented as a subset of Q:

$$R = \{U_i mastered | U_i \in Q\} \tag{3.2}$$

In addition to that, let r denote the inferred knowledge state:

$$r = \{P(U_1), P(U_2), .., P(U_n)\} \tag{3.3}$$

in which $P(U_i)$ is the probability of KU U_i, and n is the number of KU in Q. Therefore, $P(U_i)$ shows the probability that U_i is mastered, i.e. $P(U_i \in R)$. We can calculate the global mastery of the domain Q as: $\frac{\sum P(U_i)}{n}$.

The global mastery can be interpreted as *the probability that the user in question would master an arbitrarily chosen KU.* If Q represents a term exam for example, it would form an estimate of the subject's expected score over the whole test.

Now we need to discuss the strength of a relationship between nodes. Each arc in a POKS, for example $S_{i\Rightarrow j}$, has two associated weights, $W_{i\Rightarrow j}$ and $W_{\sim j\Rightarrow \sim i}$. These weights represent the "strength" of the surmise relation. The choice of estimators for these two weights depends on the inference propagation scheme. In the case of POKS, the weights $W_{i\Rightarrow j}$ and $W_{\sim j\Rightarrow \sim i}$ are represented by "odds ratios". They measure the influence that a KU has on the odds of another KU. For example, if KU A has a strong positive influence on the odds of B the ratio $\frac{O(B|A)}{O(B)}$ will be high, such that the observation that A is true will bring the updated probability of B close to one. The details of how the weights are obtained and their role in the evidence propagation scheme are given later in this chapter.

3.4.2 The induction of POKS from data

In POKS, as mentioned earlier, we focused on the induction of knowledge structures closed under union and intersection, that is, knowledge structures that can be completely represented by a partial order. We put a strong emphasis on the ability of this structure to produce correct inferences with small amounts of data (the experiments we report here are based on 149 examinee each with 20 questions, other experiments report positive results with 48 examinees and 34 questions and 40 examinees and 160 questions, Desmarais and Pu 2005).

However, the algorithm does not guarantee inducing the optimal topology of a network with respect, for example, to a minimal entropy criterion, or with respect to the maximum likelihood of a topology given a data set. Instead, we focus on the technique's ability to perform inferences with the network, not so much on its ability to recover a "true" underlying topology, such as done in Bayesian Network.

This is reasonable due to these two reasons:

- Assessing a user's knowledge state is at utmost importance rather than uncovering the domain's true knowledge structure,
- The cognitive perquisite structure is not the only factor that influences the validity of the inferences as we know a large portion of the knowledge structure's relations are probabilistic, (i.e. they are not based on a strict order from which discrete, true or false, and deterministic inferences could be performed). Apart from the topology which represents the directionality of influence among KU, the relations' weights and the evidence updating scheme are other factors that are just as important and that must be taken into account to assess the ability of a knowledge structure to infer an individual's knowledge state.

3.4.3 The POKS induction technique

The POKS technique induces, a set of binary implication relations from which we can apply *modus ponens* and *modus tollens* inferencing. This induction uses a small number of data cases. As an example, a relation such as $A \Rightarrow B$, can be explained as:

- if A is mastered, then B is mastered, and
- if B is not mastered, then A is not mastered.

These inferences determine the probability of mastery of B according to some new evidence of the mastery of A, or conversely, the probability of mastery of A given non-mastery of B. Therefore they are probabilistic. The values will depend on the surmise relation's strength, as determined by its associated weights, $W_{i \Rightarrow j}$ and $W_{\sim j \Rightarrow \sim i}$ and on the evidence propagation scheme.

In an ideal case, if there were an implication relation $A \Rightarrow B$, then we would never expect to find that someone knows A but does not know B. This assertion translates into the following two conditions:

$$P(B|A) = 1 \tag{3.4}$$

$$P(\sim A|\sim B) = 1 \tag{3.5}$$

We need to pay attention to the fact that many surmise relations do not have a strict order, in a way that the two conditional probabilities will be more or less close, but not equal, to 1. Furthermore, sampling errors has an affect on the measured conditional probabilities. Therefore we need a statistical model for implication, or surmise relation.

As it will be discussed in following sections, the statistical model behind the implication relation is based on two test of hypotheses to verify that the conditional probabilities of $P(B|A)$ and $P(\sim A|\sim B)$ are above a given minimal threshold, and a third test to verify that the conditional probabilities are different from the initial probabilities.

3.4.4 Hypothesis tests on $P(B|A)$ and $P(\sim A|\sim B)$

The two test of hypothesis over the conditional probabilities can be stated as follow:

$$P([P(B|A) \leq P_c]|D) < \alpha_c \tag{3.6}$$

$$P([P(\sim A|\sim B) \leq P_c]|D) < \alpha_c \tag{3.7}$$

where:

P_c: minimal conditional probability chosen for $P(B|A)$ and $P(\sim A|\sim B)$. This is an indicator of the strength of the knowledge structure's surmise relations.

α_c: the alpha error of the minimal conditional probability tests. This parameter determines the proportion of relations that erroneously fall below P_c.

Table 3.1. Distribution of observed co-occurrences.

	B	~B
A	$N_{A\wedge B}$	$N_{A\wedge\sim B}$
~A	$N_{\sim A\wedge B}$	$N_{\sim A\wedge\sim B}$

D: the frequency distribution of co occurrences of A and B in a data sample, as illustrated in Table 3.1, and where each value of $N_{X\wedge Y}$ corresponds to one of the following 4 conditions:

- $N_{A\wedge B}$: co occurrences of A mastered and B mastered;
- $N_{A\wedge\sim B}$: co occurrences of A mastered and B not mastered;
- $N_{\sim A\wedge B}$: co occurrences of A not mastered and B mastered;
- $N_{\sim A\wedge\sim B}$: co occurrences of A not mastered and B not mastered;

Let us demonstrate how these tests of hypothesis can be conducted. The frequency pair ($N_{A\wedge B}$, $N_{A\wedge\sim B}$) and ($N_{\sim A\wedge\sim B}$, $N_{A\wedge\sim B}$) are stochastic variables with a probability distribution that follows the binomial distribution $Bin(k,n,p)$, where:

$$k = \begin{cases} N_{A\wedge B} & \text{for the pair } (N_{A\wedge B}, N_{A\wedge\sim B}) \\ N_{\sim A\wedge\sim B} & \text{for the pair } (N_{\sim A\wedge\sim B}, N_{A\wedge\sim B}) \end{cases} \tag{3.8}$$

$$n = k + N_{A\wedge\sim B} \tag{3.9}$$

and

$$p = \begin{cases} P(B \mid A) \text{ for the pair } (N_{A\wedge B}, N_{A\wedge\sim B}) \\ P(\sim A \mid \sim B) \text{ for the pair } (N_{\sim A\wedge\sim B}, N_{A\wedge\sim B}) \end{cases} \tag{3.10}$$

In other words, the probability distribution of each frequency pair is determined by a binomial function having $P(B|A)$ or $P(\sim A|\sim B)$ as one of its argument, and by two cell

values in the distribution D. Therefore, the test of hypothesis for $A \Rightarrow B$ can be obtained by computing a lower tail confidence interval over a binomial function:

$$P(X \leq N_{A \wedge \sim B}) = \sum_{i=0}^{N_{A \wedge \sim B}} \binom{n}{i} p^{n-i}(1-p)^{i} \tag{3.11}$$

Where n has the same definition as above, and p is set to the desired minimal conditional probability, P_c.

3.4.5 Interaction test

We use the two tests of hypothesis on conditional probabilities to ensure that the minimal "strength" of the relation is above a predetermined threshold, p_c. In order to have a complete assessment, we still need to verify that the conditional probabilities are different than the no conditional probabilities, that is:

$$P(B|A) \neq P(B) \tag{3.12}$$

$$P(\sim A|\sim B) \neq P(\sim A) \tag{3.13}$$

To verify these conditions we can apply a χ^2 test on the 2 by 2 contingency table:

$$P(\chi^2) < \alpha_i \tag{3.14}$$

Where α_i is the alpha error of interaction. For small samples $(N < 50)$, the Fisher exact test should be used instead.

These three tests are sufficient to characterize a surmise relation and to ensure that

- Its "strength" is above a minimum and
- That a maximum error tolerance is set.

3.4.6 Applying the induction technique to a specific example

In this section we illustrate how the induction technique is applied to a specific example. In this example we wish to verify the existence of $A \Rightarrow B$. In the first step of implication relation induction, we compile a two dimensional contingency table for the co occurrences of A and B from an empirical data set. Table 3.2 shows a possible table of co occurrences.

Table 3.2. Example distribution of observed co-occurrences

	B	~B
A	20 ($N_{A\wedge B}$)	1 ($N_{A\wedge\sim B}$)
~A	8 ($N_{\sim A\wedge B}$)	1 ($N_{\sim A\wedge\sim B}$)

Now, in the second step of the induction method, we conduct the above-mentioned three tests of hypothesis.

For this example, we assume $P_c = 0.85$ and $\alpha_c = \alpha_i = 0.20$. Subsequently, the binomial hypothesis test for $P(B|A)$ can be computed as follows from equation 3.11:

$$\begin{aligned} &P(x \le N_{A\wedge\sim B}) = P(x \le 1) = P(x = 0) + P(x = 1) \\ &= P(x \le N_{A\wedge\sim B}) = \binom{21}{0} 0.85^{21} 0.15^{0} + \binom{21}{1} 0.85^{20} 0.15^{1} = 0.155 \end{aligned} \tag{3.15}$$

hence,

$$P(x \le N_{A\wedge\sim B}) < \alpha_c \tag{3.16}$$

where symbol $\binom{j}{k}$ represents the number of combinations of k in j. The inference with $A \Rightarrow B$ in the *modus ponens* direction is significant with confidence level $(1-\beta_c)$

In a similar way, the test for P(~A|~B) yields:

$$P(x \le N_{A\wedge\sim B}) = \binom{2}{0}0.85^2 0.15^0 + \binom{2}{1}0.85^1 0.15^1 = 0.98 \tag{3.17}$$

hence,

$$P(x \le N_{A\wedge\sim B}) \not< \alpha_c \tag{3.18}$$

As we can see, the test of $P(\sim A|\sim B)$ is not satisfied therefore $A \Rightarrow B$ cannot be used for modus tollens inference. Hence, the implication relation $A \Rightarrow B$ is rejected.

We don't bother ourselves here with the interaction test (3.14) in this example since the second test on $P(\sim A|\sim B)$ failed.

3.4.7 Estimating the implication weight

We estimate the implication weights $W_{a\Rightarrow b}$ and $W_{\sim b\Rightarrow\sim a}$, using sample data and the odds ratio as explained below:

$$W_{a\Rightarrow b} = \frac{O_{est}(B|A)}{O_{est}(B)} \tag{3.19}$$

$$W_{\sim b\Rightarrow\sim a} = \frac{O_{est}(A|\sim B)}{O_{est}(A)} \tag{3.20}$$

where $O(X)$ represents the odds of X and $O(X|Y)$ represents the odds of X given Y, that is:

$$O_{est}(X) = \frac{P_{est}(X)}{P_{est}(\sim X)} \tag{3.21}$$

$$O_{est}(X|Y) = \frac{P_{est}(X|Y)}{P_{est}(\sim X|Y)} \tag{3.22}$$

To obtain $P_{est}(B|A)$ and $P_{est}(\sim A|\sim B)$, we can use data sample using the following formula:

$$P_{est} = \frac{k+1}{n+2} \tag{3.23}$$

where:

$$n = \begin{cases} \mathrm{N_{A\^{}B}} + \mathrm{N_{A\^{}\sim B}} \text{ for } \mathrm{P(B|A)} \\ \mathrm{N_{\sim A\^{}\sim B}} + \mathrm{N_{A\^{}\sim B}} \text{ for } \mathrm{P(\sim A|\sim B)} \end{cases} \tag{3.24}$$

$$k = \begin{cases} \mathrm{N_{A\^{}B}} \text{ for } \mathrm{P(B|A)} \\ \mathrm{N_{\sim A\^{}\sim B}} \text{ for } \mathrm{P(\sim A|\sim B)} \end{cases} \tag{3.25}$$

3.4.8 Applying Bayesian inferences with POKS

Now that our knowledge structure is obtained and parameterized, the task of assessing someone's knowledge state can be done using this structure with a Bayesian induction technique in order to estimate each node's "truth value", i.e. the probability of mastery.

Every time a node is assigned a new probability, such as when mastery or failure is observed, then every other node it connects to is reassigned a new probability of mastery, and the process is repeated recursively until all paths from the originating node are followed:

- In the case where the probability of mastery is increased (i.e. observation of a success), then the implication links are followed in the forward direction,

- Where as if the probability is decreased (i.e. failure), the links are followed in the backward direction.

POKS uses a simple inference scheme based on standard Bayesian posterior probability update. However, for inferring from partial evidence, it uses an algorithm similar to the Prospector inference algorithm of Duda *et al.* (1976).

3.4.9 The posterior computation

Giarratano and Riley (1989) use the notions of likelihood of sufficiency and likelihood of necessity for updating a node's probability. Given a surmise relation $A \Rightarrow B$, these likelihood are defined respectively as:

$$LS = \frac{O(B|A)}{O(B)} \tag{3.26}$$

$$LN = \frac{O(A|\sim B)}{O(A)} \tag{3.27}$$

They correspond respectively to $W_{a \Rightarrow b}$ and $W_{\sim b \Rightarrow \sim a}$, the POKS relations' weights described in previous sections. It follows that if we know A to be true (i.e. $P(A)=1$), then the probability of B can be updated using this form of the above equation:

$$O(B|A) = LS \times O(B) \tag{3.28}$$

and conversely, if B is known false ($P(B)=0$), then:

$$O(A|\sim B) = LN \times O(A) \tag{3.29}$$

The odds ratios are obtained using the estimated probabilities explained above. Propagation of evidence actually happens by observing evidence: If we observe $P(A)=1$

with our example $A \Rightarrow B$, it is the relative effect of A on the odds of B that is propagated in the relation.

- If A has a strong effect on B, LS will be very high and it will bring the probability of B close to 1,
- Where as if A has little effect on B (i.e. if LS is close to 1) then the probability of B will only increase slightly.

An analogous relationship holds for LN when propagating backward. So far, this algorithm actually corresponds to the classic Bayesian posterior probability computation in its Odds ratio form. The Prospector algorithm departs from standard Bayesian theory in its treat of partial evidence. We return to it later.

3.4.10 Pooling and propagation of evidence

Once the POKS topology is learned from data, it can be used for knowledge inference as outlined before in this chapter. The knowledge structure inference process makes strong independence assumptions, namely that evidence variables are independent.

For example, assuming that we have n number of relations of the form $E_i \Rightarrow H$ (E for evidence and H for hypothesis), then:

$$P(E_1,...,E_n \Big| H) = \prod_i^n P(E_i|H) \tag{3.30}$$

Given the assumption of independence of equation 3.30, the probability update of H can be written in following posterior odds form:

$$O(H \Big| E_1,...,E_n) = \prod_i^n O(H|E_i) \tag{3.31}$$

where $O(H|E_i)$ represents the odds of H given evidence of E_i, and assumes the usual odds semantics $O(H|E_i) = \frac{P(H|E_i)}{1 - P(H|E_i)}$. This allows us to use Bayes' Theorem in its version based on odds and likelihood algebra:

$$O(H|E) = LS_{EH} \times O(H) \tag{3.32}$$

$$O(H|\overline{E}) = LN_{HE} \times O(H) \tag{3.33}$$

and where LS and LN are respectively the likelihood of sufficiency and the likelihood of necessity.

$$LS_{HE} = P(H|E)/P(\overline{H}|E) \tag{3.34}$$

$$LN_{HE} = P(\overline{H}|E)/P(\overline{H}|\overline{E}) \tag{3.35}$$

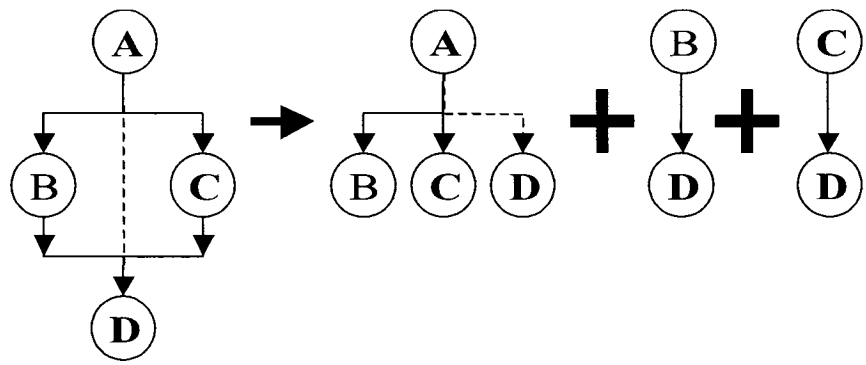


Figure 3.3. Transformation of POKS into a set of single layer networks.

Note that in the current study, we will not use transitive/recursive propagation as was performed for previous studies with POKS (Desmarais *et al.* 1996, Desmarais and Pu 2005). Instead, we rely on the fact that if we have strong surmise relations $A \Rightarrow B \Rightarrow C$,

then we would also expect to find $A \Rightarrow C$ according to the POKS structural learning algorithm. In other words, if we have $A \Rightarrow B$ and $B \Rightarrow C$, no probability update is performed over C upon the observation of A, unless a link $A \Rightarrow C$ is explicitly derived from the data.

This principle is illustrated in figure 3.3. A simple POKS topology can be transformed into three single layered networks. The dotted line would normally be derived from data if the network contains strong surmise relations. Unpublished experimental results show that the performance is very close between the two alternatives: propagation within single layer networks or recursively in POKS.

3.4.11 An example of evidence propagation in a simple knowledge structure

We use a simple knowledge structure (figure 3.4) to demonstrate evidence propagation with the Prospector scheme and to build a clearer picture of the overall system's behaviors.

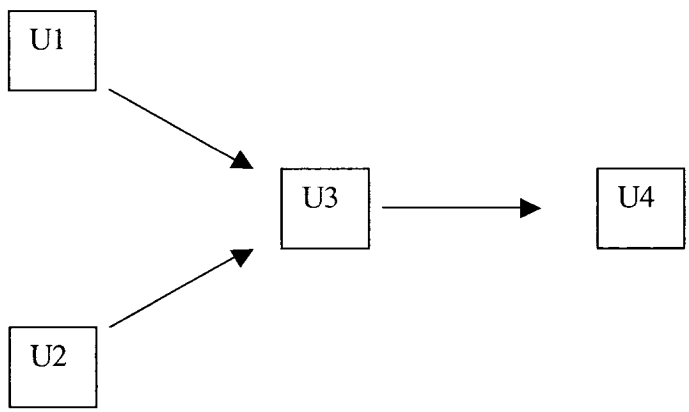


Figure 3.4. Simple knowledge structure in POKS

Figure 3.4 provides the schema contains three surmise relations. The nodes' joint distribution for each relation is reported in Table 3.3. As an example, the joint distribution for $N_{A \wedge B}$ represents total number of cases (students) who have answered

question A correctly but failed to answer question B which in our table is 0. The relations' values for *LN* and *LS* were computed over this joint distribution.

Table 3.3. Joint distributions and likelihood ratios.

Relations	Cell				Likelihood	
	$N_{A\wedge B}$	$N_{A\wedge\sim B}$	$N_{\sim A\wedge B}$	$N_{\sim A\wedge\sim B}$	LS	LN
U1⇒U3	16	0	12	36	21.53	0.075
U2⇒U3	20	4	8	32	5.32	0.245
U3⇒U4	28	0	12	24	17.95	0.051

For example, the value of *LS* for $U_1 \Rightarrow U_3$ can be obtained using equation 3.32. For the purpose of simplicity, let us use an equivalent form of equation 3.32 which involves probabilities instead of odds ratios (Giarratano and Riley 1989):

$$LS = \frac{P(A|B)}{P(A|\sim B)}$$

$$LS = \frac{(N_{A\wedge B}+1)/(N_{A\wedge B}+N_{\sim A\wedge B}+2)}{(N_{A\wedge\sim B}+1)/(N_{A\wedge\sim B}+N_{\sim A\wedge\sim B}+2)} \qquad (3.36)$$

$$LS = \frac{(16+1)/(16+12+2)}{(0+1)/(0+36+2)} = 21.53$$

The values for *LN* can be obtained in a similar manner from the joint distributions. The likelihoods *LN* and *LS* are the only parameters required by the propagation algorithm, and we can now proceed with the example to demonstrate the details of the propagation computations.

Evidence propagation occurs after evidence is observed and node's probability has changed:

- Evidence will propagate forward to the connected nodes if the change is positive,
- Whereas it will propagate backward if the change is negative.

Assuming that observation of mastery changes a node's probability to 1, and observation of no-mastery to 0, let us simulate two scenarios of evidence propagation. Table 3.4 reports the results of such a simulation when the initial probabilities of all nodes are set to 0.5. The first scenario corresponds to the successive observations of mastery of U_1 followed by U_2. The second scenario corresponds to the observation of non-mastery of U_4.

Let us examine in detail the computation involved in the very first observation (U_1) in order to see how these results are obtained: According to equation 3.32 we have:

$$O(U_3|U_1) = LS \times O(U_3)$$
$$= 21.53 \times \frac{0.5}{1-0.5} = 21.53 \tag{3.37}$$

From the general formula

$$P(X) = \frac{O(X)}{O(X)+1} \tag{3.38}$$

We can drive $P(U_3|U_1)$ from $O(U_3|U_1)$ as follows:

$$P(U_3|U_1) = \frac{21.53}{21.53+1} = 0.956 \tag{3.39}$$

Table 3.4. Scenarios of evidence propagation

Relations	Probability			
	U_1	U_2	U_3	U_4
Scenario 1:				
Initial	0.5	0.5	0.5	0.5
U1	1	0.5	0.956	0.908
U2	1	1	0.991	0.977
Scenario 2:				
Initial	0.5	0.5	0.5	0.5
U1	0.111	0.226	0.048	0

Using the Prospector scheme to propagate partial evidence, the propagation continues forward from node U_3 to node U_4. According to the propagation scheme, this process consists in estimating $P(U_4|U_1)$ based on a linear interpolation between $P(U_4)$ and $P(U_4|U_3)$. The updated value is in part a function of the amount of change in $P(U_3)$ induced by the observation of U_1. It is computed as follows:

$$P(U_4|U_1) = P(U_4) + \frac{P(U_3|U_1) - P(U_3)}{1 - P(U_3)} = [P(U_4|U_3) - P(U_4)] \qquad (3.40)$$

All values in this formula are known with the exception of $P(U_4|U_3)$ which is derived in the same manner as $P(U_3|U_1)$ was derived:

$$\begin{aligned} O(U_4|U_3) &= LS \times O(U_4) \\ &= 17.95 \times \frac{0.5}{1 - 0.5} = 17.95 \end{aligned} \qquad (3.41)$$

However, and as mentioned earlier, we do not apply this partial evidence propagation algorithm for the current study, contrary to the POKS framework described in Desmarais *et al.* (1996) and Desmarais and Pu (2005).

3.5 Conclusion

Now that we have explored the theory behind POKS, we will turn our attention to the ability of the POKS framework to assess knowledge with an experimental simulation methodology. But before that we need to look at the standard Bayesian network using DAG with knowledge engineering. This will help us compare the POKS framework to the Bayesian network approach at different levels: POKS, Bayesian Network and combination of Bayesian network and the POKS approach by introducing Concept Nodes in POKS. We look at the possibilities that offer the potential to improve the knowledge

assessment process. This improvement will be verified using an imperial study for both methods.

CHAPTER 4. KNOWLEDGE ASSESSMENT IN STANDARD BAYESIAN NETWORK

This chapter discusses Bayesian network application in educational assessment. We later take the BN developed by Vomlel (2004) as the basis for comparing POKS with a Bayesian network approach. We start with an introduction to explain basic definitions such as skills, abilities, and misconceptions used in student modeling and how they are used in graphical models. Then we explain how this model is build and finally we go through the design of the test and experiments.

4.1 Introduction

In order to design an assessment test and diagnose the presence or absence of a person's skills, an educational test designer specifies a set of ***tested skills***, ***abilities***, ***misconceptions***, etc. and a bank of questions, tasks, etc. Let assume $\mathbf{S} = \{S_1,...,S_k\}$ provides the set of tested skills, abilities, misconceptions, etc. On the other hand $\mathbf{X} = \{X_1,...,X_k\}$ denote the bank of questions, tasks, etc. This is the job of test designer to specify which skills are directly related to each question. These relations are often probabilistic, especially if a multiple-choice test is used.

An approach used in test design can be to construct a test that consists of a fixed sequence of questions covering all tested skills. This type of test is called a ***fixed test***. Another approach (briefly mentioned in chapter one) aims at constructing an optimal test for each examinee. After each response on a question the system selects next question based on the answers of the previous questions. Since this approach requires computers for the test administration it is often referred to as computerized adaptive testing (CAT). Tests that are automatically tailored to the level of the individual examinees will be referred to as ***adaptive tests***.

Almond and Mislevy (1999) have proposed to use of ***graphical models*** for CAT. Every skill S_i and question X_j are represented by a random variable having a finite sets of values S_i and X_j, respectively. **S** is used here to denote the random variable $(S_1,...,S_k)$. Similarly, **X** will denote the random variable $(X_1,...,X_m)$. The model of Almond and Mislevy consists of one ***student model*** $P(\mathbf{S})$ and one ***evidence model*** $P(\mathbf{X}|\mathbf{S})$ for each question **X**. The student model describes relations between skills by use of a joint probability distribution $P(\mathbf{S})$ defined on the variables of the student model. Using the approach of Almond and Mislevy (1999), Vomlel (2004) has defined the overall probabilistic model of the problem as a Bayesian network

$$P(\mathbf{S},\mathbf{X}) = P(\mathbf{S}) \cdot \prod_{X_j \in X} P(X_j | S_j) \tag{4.1}$$

The approach presented by Vomlel describes an algorithm to construct a student model. Followings will be a brief explanation of his method and the design of the test on a student and evidence models for testing basic operations with fractions. This is an attempt to describe the process of learning models and present results of experiments performed with tests built using the learned models. We invite readers to refer to Vomlel (2004) for a detailed analysis of the approach.

4.2 Building models

Vomlel has used a probabilistic model $P(\mathbf{S},\mathbf{X})$ represented by a Bayesian network to model the educational assessment problem. Lets assume we have an already collected data as $D = \{(X_1,S_1),...,(X_n,S_n)\}$. There are different methods available for structural learning of Bayesian networks models from collected data. Generally there are two classes of the structural learning algorithms:

(1) Score-based and

(2) Constraint-based learning algorithms.

Algorithms in both classes perform an informed search through the search space of all possible models guided by a search strategy. Vomlel aim's at correct prediction of absence or presence of skills S. Therefore a score used to evaluate different models could be the conditional log-likelihood (CLL) of a model P given data D.

$$CLL(P|D) = \sum_{i=1}^{n} \log(P(S^i|X^i)). \tag{4.2}$$

One fundamental problem with this score is that in order to maximize CLL for model structure it is necessary to search over the whole space of model parameters. This makes the method computationally expensive.

Experiments of Cheng and Greiner (1999) suggest that learning algorithms based on series of conditional independence test provides Bayesian network classifiers that perform well. Vomlel has used a constraint-based learning algorithm – the Hugin PC algorithm - a variant of the original PC algorithm of Spirtes *et al.* (1993) to learn the structure of the student model $P(\mathbf{S})$. This algorithm is based on series of conditional independence tests.

Next, we briefly describe the steps of the PC algorithm. A more detailed description of the Hugin implementation of the PC algorithm can be found in a paper by Jensen *et al.* (2002). The algorithm performs following steps:

(1) Statistical tests for conditional independence between all pairs of variables are performed.

(2) An undirected link is added between each pair of variables for which no conditional independence was found.

(3) A collider is a pair of links directed such that they meet head-to-head in a node. For example, $A \Rightarrow B \Leftarrow C$ is a Collider. Colliders are then identified, ensuring that no directed cycles occur. E.g., if variables A and B are found to be dependent, variables B and C are found to be dependent, but variables A and C are found to be

conditionally independent given a set of variables not containing B, then this can be represented by the collider structure $A \Rightarrow B \Leftarrow C$.

(4) Links whose direction can be derived from the identified conditional independences and colliders are directed.

(5) The remaining undirected links are directed randomly, so that no directed cycle occurs.

The structure produced by PC algorithm is under the assumptions of infinite data sets, perfect tests, and DAG faithfulness (DAG faithfulness means that the data can be assumed to be simulated from a probability distribution that factorizes according to a DAG.). In the case of limited data sets, however, these algorithms often derive too many conditional independence statements. Also, they may in some cases leave out important dependence relations. Thus, *the learned structure should be inspected by a domain expert*. Vomlel combined the Hugin PC algorithm applied to collected data with an expert knowledge of the modeled domain.

4.3 Model for basic operations with fractions

Vomlel built a Bayesian network from data over an arithmetic test. We describe the model of the skills behind this test and the types of items found.

4.3.1 Student Model

The learning process that resulted in a student model consisted of several steps. First, a group of students from Aalborg University prepared paper tests that were given to students at Brønderslev High School. Four elementary skills, four operational skills, and abilities to apply operational skills to complex tasks were tested. Table 4.1 presents elementary and operational skills.

Table 4.1. Elementary and operational skills.

Label	Description	Example
CP	Comparison (common numerator or denominator)	$\frac{1}{2}>\frac{1}{3},\frac{2}{3}>\frac{1}{3}$
AD	Addition (common denominator)	$\frac{1}{7}+\frac{2}{7}=\frac{1+2}{7}=\frac{3}{7}$
SB	Subtraction (common denominator)	$\frac{2}{5}-\frac{1}{5}=\frac{2-1}{5}=\frac{1}{5}$
MT	Multiplication	$\frac{1}{2}\cdot\frac{3}{5}=\frac{3}{10}$
CD	Finding common denominator	$(\frac{1}{2},\frac{2}{3})=(\frac{3}{6},\frac{4}{6})$
CL	Canceling out	$\frac{4}{6}=\frac{2\cdot 2}{2\cdot 3}=\frac{2}{3}$
CIM	Conversion to mix numbers	$\frac{7}{2}=\frac{3\cdot 2+1}{2}=3\frac{1}{2}$
CMI	Conversion to improper fractions	$3\frac{1}{2}=\frac{3\cdot 2+1}{2}=\frac{7}{2}$

Totally 149 students solved the test. The university students analyzed the tests and summarized the results. During this phase, seven types of misconception were discovered. Table 3.2 presents misconceptions observed in Brønderslev High School.

4.3.2 Evidence models

For each task or question an evidence model is created. Tasks are simple arithmetic calculations as shown below:

$$
\begin{aligned}
T_1 &\quad (\frac{3}{4}\cdot\frac{5}{6})-\frac{1}{8}=\frac{15}{24}-\frac{1}{8}=\frac{5}{8}-\frac{1}{8}=\frac{4}{8}=\frac{1}{2} \\
T_2 &\quad \frac{1}{3}-\frac{1}{12}=\frac{4}{12}-\frac{1}{12}=\frac{3}{12}=\frac{1}{4} \\
T_3 &\quad \frac{1}{4}\cdot 1\frac{1}{2}=\frac{1}{4}\cdot\frac{3}{2}=\frac{3}{8} \\
T_4 &\quad (\frac{1}{2}\cdot\frac{1}{2}).(\frac{1}{3}+\frac{1}{3})=\frac{1}{4}\cdot\frac{2}{3}=\frac{2}{12}=\frac{1}{6}
\end{aligned}
\qquad (4.3)
$$

An example of a student model is given in Figure 4.2. Misconceptions happened for students during the calculations and used in the student model are explained in table 4.2.

Table 4.2. Misconceptions

Label	Description	Occurrence
MAD	$(\frac{a}{b}+\frac{c}{d})=\frac{a+c}{b+d}$	14.8%
MSB	$(\frac{a}{b}-\frac{c}{d})=\frac{a-c}{b-d}$	9.4%
MMT1	$(\frac{a}{b}\times\frac{c}{b})=\frac{a\times c}{b}$	14.1%
MMT2	$(\frac{a}{b}\times\frac{c}{b})=\frac{a+c}{b\times b}$	8.1%
MMT3	$(\frac{a}{b}\times\frac{c}{d})=\frac{a\times d}{b\times c}$	15.4%
MMT4	$(\frac{a}{b}\times\frac{c}{d})=\frac{a\times c}{b+d}$	8.1%
MC	$a\frac{b}{c}=\frac{a\times b}{c}$	4.0%

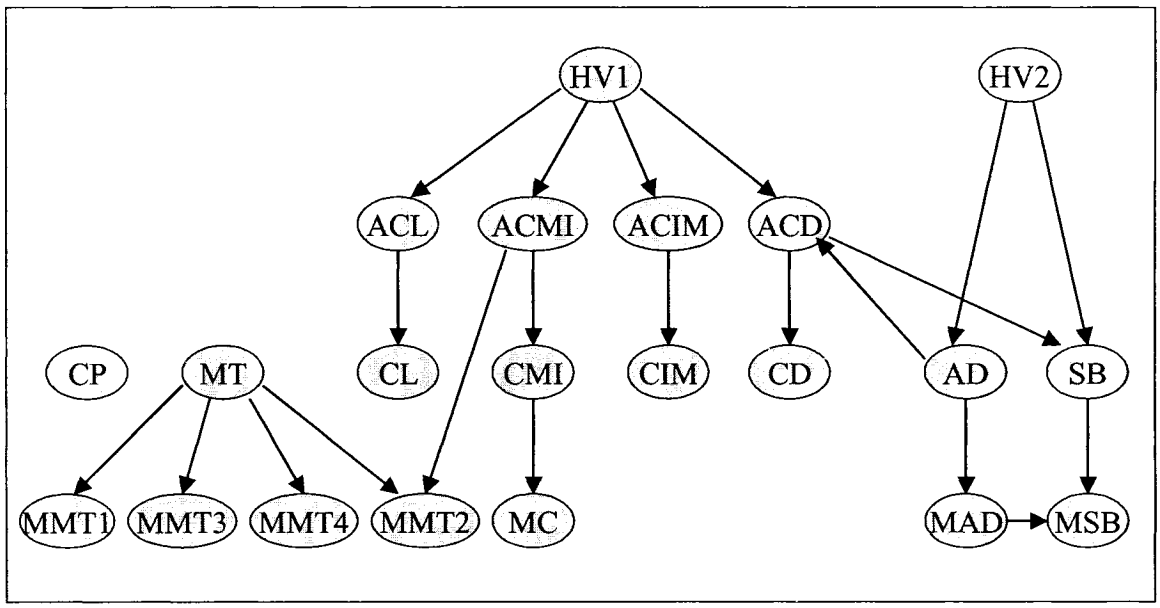


Figure 4.1. Student model describing relations between skills and misconceptions.

Lets assume that a student is able to solve certain tasks if and only if she has the necessary skills and does not have certain misconceptions. Thus, The tasks can formally be described as T_1, T_2, T_3, and T_4 by logical formulas:

$$
\begin{aligned}
T_1 &\Leftrightarrow \text{MT} \,\&\, \text{CL} \,\&\, \text{ACL} \,\&\, \text{SB} \,\&\, \sim \text{MMT3} \,\&\, \sim \text{MMT4} \,\&\, \sim \text{MSB} \\
T_2 &\Leftrightarrow \text{SB} \,\&\, \text{CL} \,\&\, \text{ACL} \,\&\, \text{CD} \,\&\, A\text{CD3} \sim \text{MSB} \\
T_3 &\Leftrightarrow \text{MT} \,\&\, \text{CMI} \,\&\, \text{ACMI} \,\&\, \sim \text{MMT3} \,\&\, \sim \text{MMT4} \,\&\, \sim \text{MC} \\
T_4 &\Leftrightarrow \text{MT} \,\&\, \text{AD} \,\&\, \text{CL} \,\&\, \text{ACL} \,\&\, \sim \text{MMT1} \,\&\, \sim \text{MMT2} \,\&\, \sim \text{MMT3} \,\&\, \sim \text{MMT4} \,\&\, \sim \text{MAD}
\end{aligned}
\tag{4.4}
$$

The assumption of deterministic relations between skills and the actual outcome of a task might be an unrealistic one. A student can make a mistake even if she has all abilities needed to solve a given task. On the other hand, a correct answer does not necessarily mean that the student has all abilities since she may guess the right answer, e.g., in a test where she is to select one answer from a given set of answers. Vomlel has used a task variable T_i as the ability to solve the corresponding task and a non-deterministic model for the description of the dependence between the skill T_i and the actual outcome of the

corresponding task X_i. Using this assumption he modeled "guessing" using conditional probability $P(X_i|\sim T_i)$ and "mistakes" using $P(\sim X_i|T_i)$. (Figure 4.3)

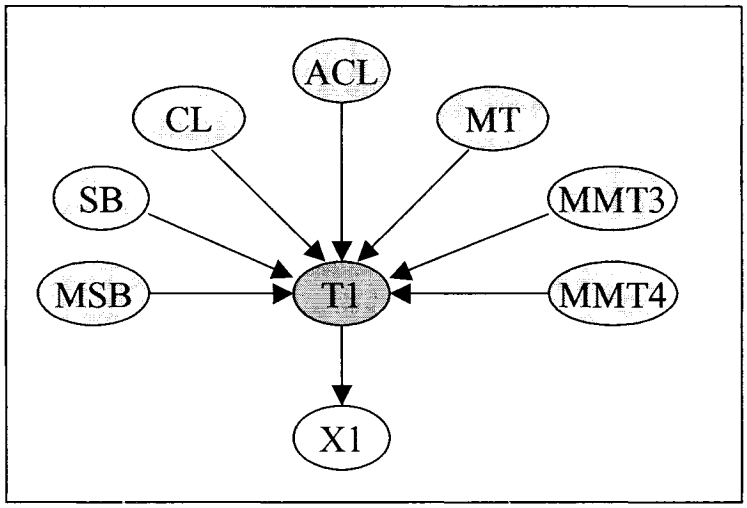


Figure 4.2 Evidence Model of task T_1

4.4 Conclusion

In this chapter we explored briefly an example of Bayesian network approach to educational assessment. Next chapter will explain a comparison between two previously explained models in chapters 3 and 4 using the same empirical data provided by Vomlel (2004). We analyze the issues between the choice of knowledge engineered vs. learned student models and present a framework based on observable item to item structures that aims towards providing fine grained learned student models. We compare that approach with a Bayesian network framework in an experimental simulation with data from pupils on an arithmetic test.

CHAPTER 5. SIMULATIONS AND RESULTS

In this chapter we investigate the ability of the POKS framework, standard Bayesian network and combination of two, to accurately assess knowledge using an experimental simulation methodology. The POKS framework is compared to a Bayesian network approach at two levels:

1. Question predictive accuracy (IPA): To evaluate the performance for predicting actual responses;
2. Concept predictive accuracy (CPA): To evaluate the performance for assessing concept mastery;

The metrics are calculated during the simulation by rounding the calculated probabilities to 1 for $P(X) \geq 0.5$ (Mastery of the subject) and 0 for $P(X) < 0.5$ (non-Mastery of the subject) and using following formula:

$$IPA = \frac{\sum^{s}\sum^{i}\|P(X_{si})\|}{S * i} \tag{5.1}$$

$$IPC = \frac{\sum^{s}\sum^{c}\|P(X_{sc})\|}{S * c} \tag{5.2}$$

In this formula s is the number of subjects, i is the number of items and c is the number of concepts.

For the purpose of comparison and see the effect of this technique we use Vomlel's (2004) data. We look at the analysis study of the performance of a number of Bayesian network models over these two dimensions as a comparison point (Vomlel, 2004).

5.1 Experimental data

Vomlel experimented with a number of BN models to determine each model's ability to predict the actual question item success and concept mastery of 149 pupils who completed a 20 question items test of basic fraction arithmetic. The structural details and the decomposition method of this test are explained in chapter 4. A total of 9 models were tested and we report the results of the best performing one and which corresponds to the model in figure 5.1. For assessing the mastery of concepts, Vomlel used an independent source: expert judgment on the mastery of each concept based on the specific answer pattern to each 20 questions items. That data allows the training of the BN and POKS frameworks. This situation is atypical, since we generally do not have the luxury of "observing" concept mastery and training a model with such data, but it conveniently allows us to do an experimental comparison of the two approaches.

In the first simulation we use the exact replica of the knowledge representation developed by Vomlel as shown in figure 5.1 The only difference is that for simplicity, we have omitted the implementation of intermediate nodes tasks and effects of the conceptions and misconceptions are directly transferred to children (questions and answers in our case, X_i)

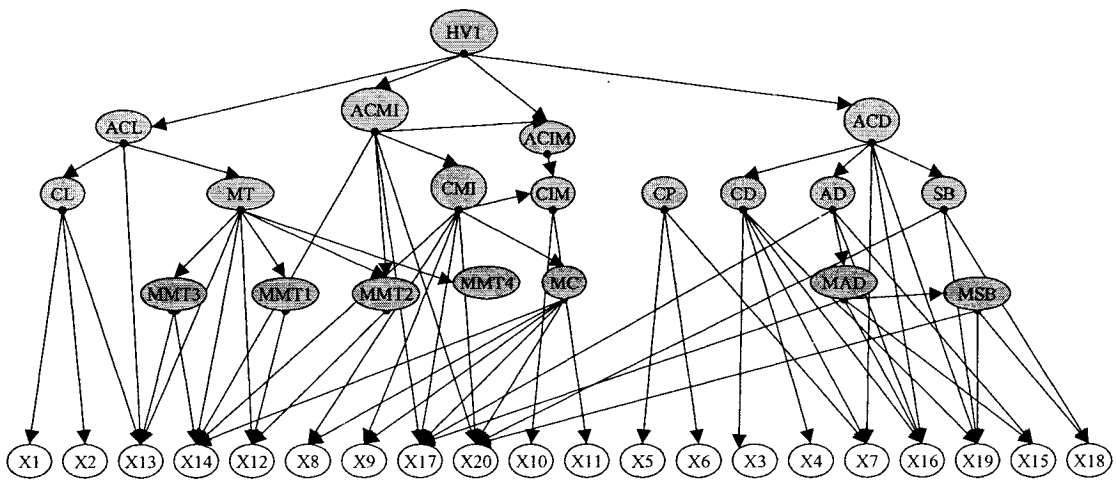


Figure 5.1. Knowledge structure with the misconceptions

The simulation data contained the values of all the concept (11), misconception (7) and question nodes (20).

In another simulation we used the same data except that we removed the misconceptions from the structure. The structure is presented in Figure 5.2.

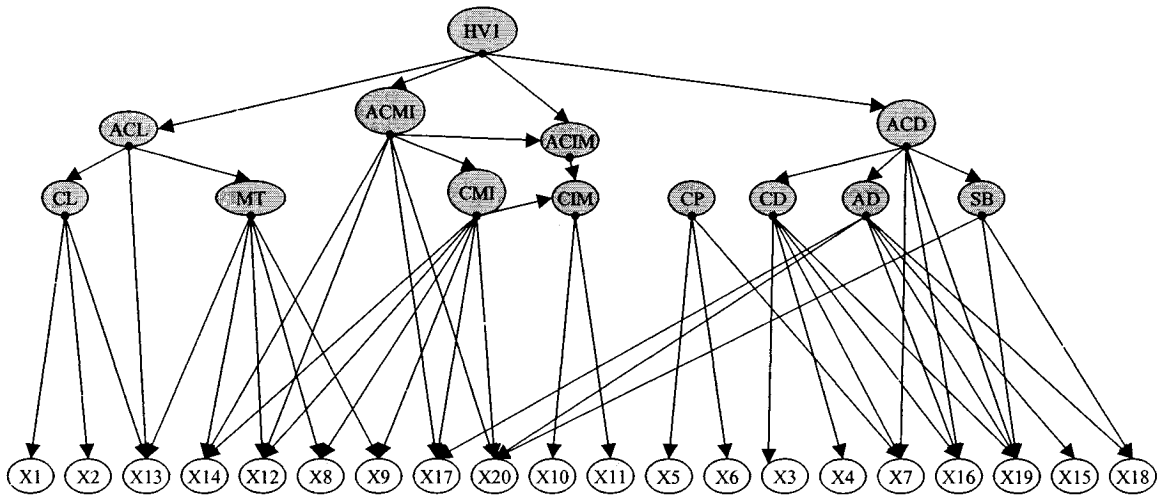


Figure 5.2. Knowledge structure without the misconceptions

5.2 Simulation development environment

POKS has been developed using C language with standard libraries and can be used under Unix and Linux operating systems. In order to compare POKS framework and standard Bayesian networks using same simulation data, we explored different available modeling and implementation for Bayesian network software. Literature provides many development environment and application, each presenting different advantages and disadvantages. A comparison of these packages can be found in Appendix I. Vomlel used Hugin Decision Engine through Hugin Development Environment (Hugin ver 6.0) to construct the knowledge structure and apply the Bayes inference accordingly. To replicate his idea for the purpose of comparison with POKS, we explored Hugin's engine and some other available Bayesian inference packages. Following is the result of our investigation, which resulted in picking Matlab Bayes toolbox.

5.2.1 The Hugin Development Environment Hugin

The Hugin Development Environment provides a set of tools for constructing model-based decision support systems in domains characterized by inherent uncertainty. The models supported are Bayesian networks (BNs) and their extension influence diagrams (IDs). The Hugin Development Environment allows user to define both discrete domain variables and to some extent continuous domain variables in their models.

Hugin Decision Engine (HDE) can be used through the Hugin Graphical User Interface. User can also use the HDE through one of several APIs (Application Program Interfaces) which come as libraries for C, C++, and Java, and as an ActiveX server. More information can be found in their website: www.hugin.com

Due to the fact that the implementation of the Bayesian network for data supplied by Vomlel was done through Hugin development environment, therefore Hugin was the first package to use. In order to construct our knowledge structure, it was required to create a Network of 40 nodes but unfortunately due to the limitation of the Lite version in number nodes (maximum 25) we were unable to create such a structure representative of our simulation data. Therefore we decided to examine other packages.

5.2.2 MSBNx

MSBNx is a component-based Windows application (Microsoft Windows software application) for creating, assessing, and evaluating Bayesian networks. Its components can be integrated into programs, which allow them to leverage inference and decision making under uncertainty. Each model is represented as a graph or diagram. It has the capability of Graphical Editing of Bayesian Networks, it's engine contains standard assessment of probabilities and calculate exact probabilities. It provides VB as API and allowes users to use XML as format.

Unfortunately MSBNx is also not capable of handling complex structure and due to this problem, we decided to investigate other possibilities.

5.2.3 Java Bayes

JavaBayes is a system that handles Bayesian networks. It calculates marginal probabilities and expectations, produces explanations, performs robustness analysis, and allows the user to import, create, modify and export networks.

JavaBayes is the first full implementation of Bayesian networks in Java. Advantages can be, portability to different operating systems and browsers (Java) providing simple tool to reason about uncertainty in the domain of interest and finally, Java is a good object-oriented language and has a set of widgets that allow researchers to quickly prototype interfaces, and use functionality for multi-threaded processing (It is very useful for future parallel execution of inference algorithms).

The JavaBayes system is a set of tools for inferences with graphical models, containing a graphical editor, a core inference engine and a parser. JavaBayes handles Bayesian networks and produces posterior marginals, posterior expectations.

This software was rejected due to its limited computational capabilities. Users in the web based message group were complaining about this issue and found it suitable for only small structures.

5.2.4 Bayes Net Toolbox (BNT)

The Bayes Net Toolbox (BNT) is an open-source Matlab package for directed graphical models. BNT supports many kinds of nodes (Probability distributions), exact and approximate inference, parameter and structure learning, and static and dynamic models. It is a general tool for graphical model, which permits the user to write its program in Matlab, and uses the powerful capabilities of Matlab development environment to develop codes.

BNT is used in teaching and academic environment and is a good candidate as a simulation package for our case.

- Pros
    - Well known and good IDE (Interactive Development Environment)
    - A vast library of numerical algorithms and data visualization.
    - High level code and easy to understand and read (e.g., Kalman filter in 5 lines of code)
    - Matlab is the language of engineers and very and fast in development and easy to debug.
- Cons
    - Possible execution problem (out of memory error)
    - Slow (interpreter approach rather than compiler approach)
    - Commercial license is expensive.
    - Poor support for complex data structures

Based on the above-mentioned advantages and due to its rapid prototype development capabilities and the fact that it was widely used in academic environment we selected Matlab Bayes Net as a development environment. Appendix II is dedicated to an example of using inference engine in BNT.

5.3 Simulation

Before explaining the simulation methodology we clarify the method of picking the question of choice and also the concept nodes treatments in POKS compare to Bayesian Network.

5.3.1 Question of choice and entropy

If we consider computer adaptive testing, the goal of item selection is, using the least number of items to identify the examinee's ability level with maximum precision, or in

other words, choosing the most informative item. For both the BN and POKS approaches, the order of questions is adaptive and determined by entropy minimization.

The entropy of a single item X_i is defined by the usual formula:

$$H(X_i) = -[P(X_i)\log(P(X_i)) + Q(X_i)\log(Q(X_i))] \tag{5.3}$$

where $Q(X) = 1 - P(X)$.

The entropy of the whole test is the sum of all individual items' entropy:

$$H_T = \sum_i^k H(X_i) \tag{5.4}$$

If all items' probability is close to 0 or 1, the value of H_T will be small and there will be little uncertainty about the examinee's ability score. We minimize uncertainty by choosing the item with the lowest expected value of test entropy. This value is given by:

$$E_i(H_T') = P(X_i)H_T'(X_i = 1) + Q(X_i)H_T'(X_i = 0) \tag{5.5}$$

where $H_T'(X_i = 1)$ is the entropy after the examinee answers correctly to item i and $H_T'(X_i = 0)$ is the entropy after a wrong answer.

5.3.2 Concept Nodes in POKS

The POKS framework builds relations among observable question items. Therefore, the network contains no hidden nodes such as concepts and misconceptions. However, to infer mastery of concepts from observable item nodes, we need to add links from question items to concepts. In this work we used a simple approach to this problem, which aims towards automation.

An atypical approach to introducing concept nodes in POKS is to independently assess concept mastery and use a classification technique to link responses to concepts. We used a series of feed-forward single hidden node neural nets trained with independently assessed concepts as output nodes for linking input question items. For each concept node in figure 5.1, a neural net is trained with the observable question items linked to concept nodes. For example, concept node CL (left side of figure 5.1) is linked to question items X1, X2, and X13 (figure 5.3). The model's parameters are estimated from Vomlel's data again, where concept nodes mastery was independently assessed by experts.

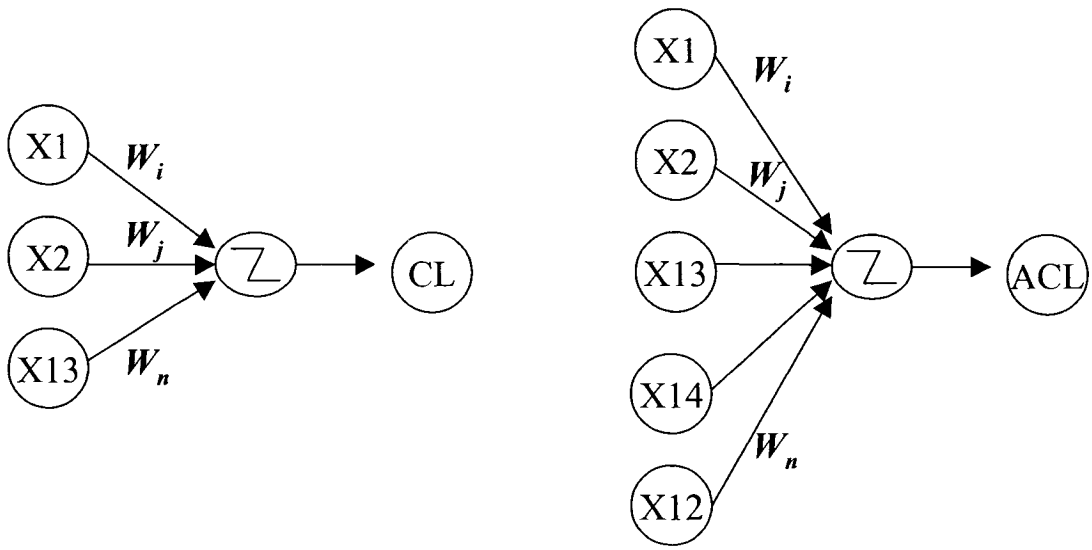


Figure 5.3. Neural net example.

We used R's "nnet" package to build the neural nets (Venables, Smith, and the R Development Core Team, 2004). We emphasize that this is not an approach that can be applied in practice, but it serves as a validation purpose here.

5.3.3 Experimental methodology

For each of the question and concept predictive experiments, we simulate the question answering process with the real subjects. The choice of the question to ask is determined by the entropy reduction optimization algorithm (as discussed in section 5.3.1) for each

of the BN and POKS frameworks. Picking the next question to pose corresponds to the one that reduces the global test entropy. Generally, it will choose the items that are most likely to significantly affect (positively or negatively) other items, which have a probability of success around 0.5, given the current state of ability assessment.

Moreover, each approach is trained over all subjects except one: the subject used in the simulation. This simulation method allows the use of N −1 data cases, while avoiding the bias in using the same data for training and validation. It implies that calibration of POKS and the BN frameworks are repeated for each subject.

Similar to (Vomlel, 2004) who used the independent concept mastery assessment data to calibrate the conditional probabilities between items and concepts, we also use this data to calibrate the links between items and concepts. The concepts themselves become observable nodes for the training phase. Single hidden node neural networks are used to link items to concepts.

The Item selection algorithm for POKS and Bayesian network is calculated as explained in section 5.3.1. This item selection is performed for each applicant and the result is used as a sequence of observing the evidence (items) in BNT.

The original data includes data for 149 subjects (Examinee) for 20 items (questions) and 20 concepts, misconceptions and hidden nodes.

5.4 Results

As mentioned before, POKS framework is compared to a Bayesian Network approach at two levels: Question predictive accuracy and concept predictive accuracy. Results are explained below.

5.4.1 Question predictive accuracy

Figure 5.4 presents the performance of both the POKS and the BN approaches for predicting the question item successes. Curves are averages of the 149 data cases as reported by (Vomlel, 2004). A third curve (dotted line) provides the performance of a non adaptive, fixed question sequence where all examinees get the same question sequence, regardless of their previous answers. In addition, no inference is performed either with POKS or the BN. The fixed sequence orders items based on the question items' initial entropies: it starts with items whose average success rate is closest to 0.5 and finishes with items whose success rate is closest to 0 or 1. It serves as a comparison baseline where only question order is optimized and no inference is performed.

The simulation results show that the POKS technique is able to predict answers to questions a few percentage point more accurately than the BN. It is a relatively small gain, but it is systematic and more significant after the fifth question, especially relative to the number of non-observed items left. On the contrary the BN's performance is very close to the fixed question sequence, showing only marginal gains for some items. The results indicates that the ability of the BN to predict question outcome is apparently not better than the simple strategy of asking the most uncertain question items first.

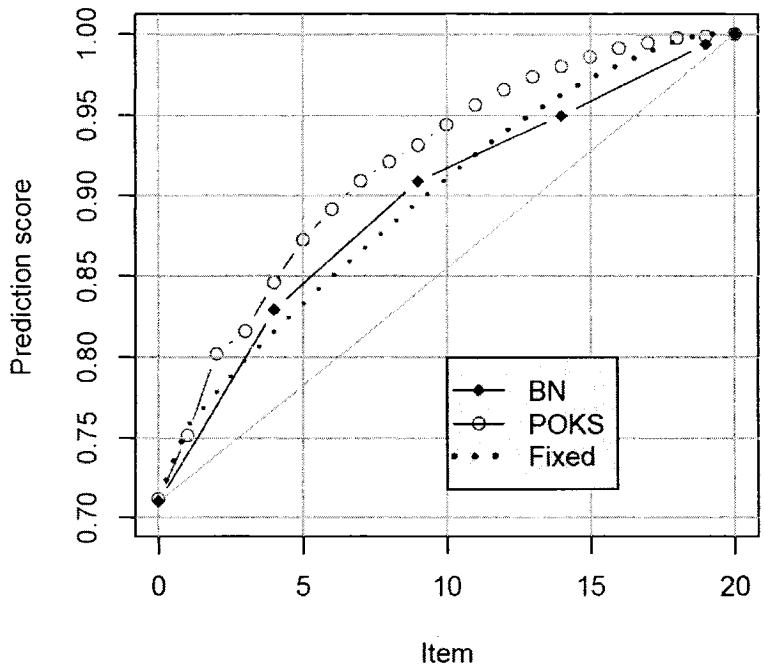


Figure 5.4. Question predictive accuracy

5.4.2 Concept predictive accuracy

Figure 5.5 shows the performance of each framework for predicting concept mastery. As we know that mastery of concepts was assessed independently by experts. During the simulations, Concepts nodes in the network, are hidden nodes but considered "observed" data for the training phase of the network.

Consequently prediction does not reach 100% accuracy as it does in the question predictive simulation. For this experiment, the performance is reversed in favor of the BN model. The BN approach quickly reaches from 74% correct to about 90% correct in only 5 items observed, and it stabilizes at close to 92% after a couple more items. The POKS performance is about 2–3% below that of the BN approach between the second and the 15th question item, and closer beyond this interval. We also note that POKS is slightly weaker (about 1%) after all 20 items are administered, indicating that the neural network model used does not perform as well as the BN for predicting hidden nodes given the full response vector.

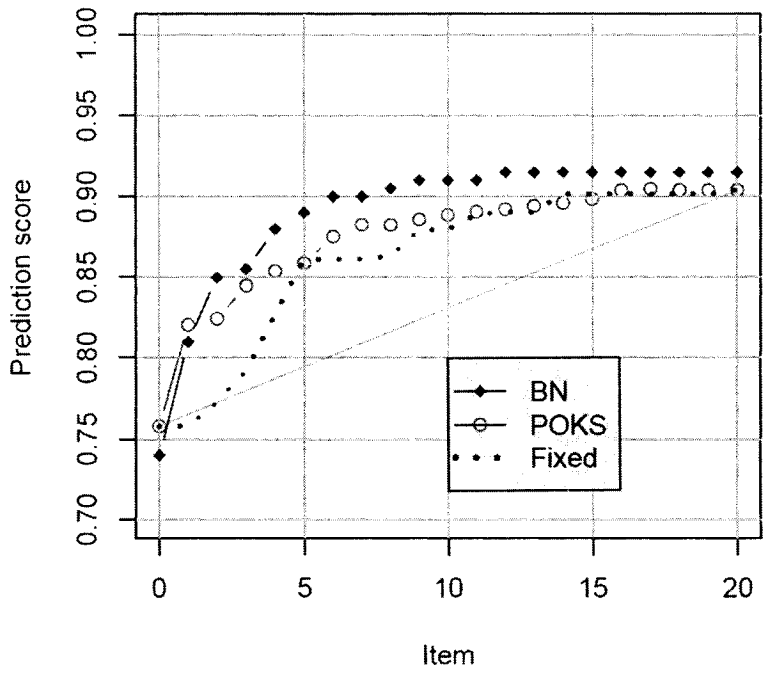


Figure 5.5. Concept predictive accuracy

The fixed item sequence corresponds to the same sequence as the one in figure 5.4. In that condition, only the neural network inference is performed. Figure 5.5's results indicates that POKS gains over this condition only until the first 10 items are posed, after which the two are very similar and close to their maximum performance.

A partial explanation for these findings is that the computation of entropy for the BN network is based on concept and item nodes, whereas it is solely based on items for POKS. It could be that if the choice of items for the BN was optimized towards items only, it could perform as well as POKS.

5.4.3 Combination of POKS with BN

Another experiment was performed by other member of group to develop the EM algorithm for POKS. This was to verify the possibility of improving over both methods by combining them together. The method of combining the approaches and the results of a simulation is explained below. Figure 5.6 describes this method.

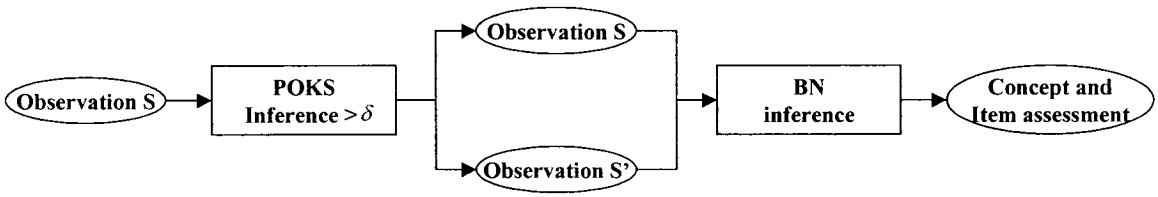


Figure 5.6. Combination algorithm of POKS with BN.

In order to combine POKS and the BN approaches, we use POKS as a first filter to augment the actual observations. If we call set of observed responses S, POKS infers a set of additional responses, *S'*. The original set, *S*, is augmented by the inferences from POKS, *S'* and the union of *S* and *S'* is considered as the new set of evidence fed to the BN. This process is repeated for every new observation, from 0 to all 20 items.

We use a threshold δ, to determine that an item is considered inferred by POKS. Every item for which the probability of mastery of POKS is greater than $1-\delta$ is considered mastered, whereas items with a probability smaller than δ is considered non mastered.

We used BNT (as mentioned in 5.2.4) to replicate Vomlel's experiment and apply this algorithm. BN structure is specified according to figure 5.1 and the conditional probabilities are determined through the EM algorithm, akin to Vomlel (2004). The junction-tree inference algorithm is used in both our experiment and Vomlel's.

We ran the simulation with POKS and the BNT package to apply the combination algorithm. We used a threshold of 0.1 for this experiment. The results are shown in figure 5.7.

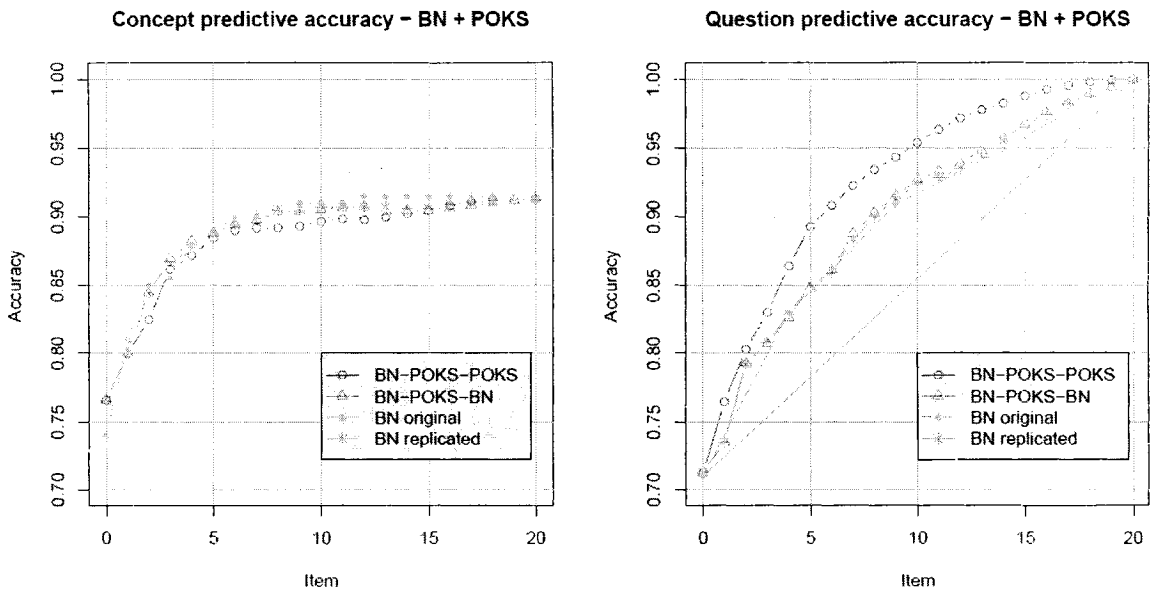


Figure 5.7. Predictive accuracy for combination of POKS and Bayesian network

The data curves reported are labeled as:

- BN-POKS-POKS: the BN performance augmented by POKS's inferences using the item selection algorithm based on POKS item nodes entropy.
- BN-POKS-BN: same as BN-POKS-POKS except that the item selection algorithm is based on the BN item and concept nodes entropy.
- BN {original/replicated}: the performance of the BN, without POKS's augmented observations, for both Vomlel's original results and for our replicated results. As expected, both curves are relatively similar since we use the same generic algorithms, although implementation details can explain the small difference.

The outcome of the simulation presented in the graphs reveals that all conditions are relatively similar. Only one curve differs significantly from the others, the BN-POKS-POKS for the question predictive results. That condition corresponds to the case where POKS inferences are combined with the BN inferences and the POKS item selection algorithm. These results indicate that the additional inferences from POKS only contributes to improve the performance of item prediction when the choice of question

item is based on POKS's expected items entropy reduction. The BN-POKS-POKS results are very close to those of POKS in figure 5.4, but they do display an improvement of about 2% between 5 and 10 observed items. This improvement indicates that the BN contributed to improve over POKS original's item prediction accuracy.

The improvement for item prediction does not transfer to concept prediction. The lack of improvement for concepts prediction is somewhat unexpected, but it might be explained by the choice of items strategy. When POKS's item choice strategy is used, it is geared to optimizing the items subset entropy and may not necessarily improve concept predictions accuracy. On the contrary, when the BN item strategy is used, it may result in fewer "augmented" inferences fed to the BN, thereby reducing the impact at concept level predictions. Another explanation has to do with the redundancy of the information inferred by both schemes. It might be that the information added by POKS's augmented set is simply redundant with the information inferred by the BN.

CHAPTER 6. DISCUSSION AND CONCLUSION

The results from this study indicate a clear advantage of each framework in question predictive accuracy and concept predictive accuracy. Here we discuss each advantage below:

- POKS is more accurate at predicting question item outcome. Based on the simulation results, POKS technique is able to predict answers to questions a few percentage point more accurately than the BN. Although this is a relatively small gain, but it is systematic gain. The BN's performance is very close to the fixed question sequence with only marginal gains for some items.
- BN is more accurate at predicting concept mastery. In this experiment, the performance of the BN model is better than POKS and quickly reaches to about 90% correct. The POKS performance is on average about 2–3% below that of the BN approach. POKS is slightly weaker (about 1%) after all 20 items are administered, showing that the neural network model used does not perform as well as the BN for predicting hidden nodes given the full response vector.

6.1 Discussion

The idea of modeling the concepts interdependencies with a BN and implementing it for knowledge assessment is not new (demonstrated by Vomlel 2004). However, this study suggests that an item-to-item structure that does not explicitly model interdependencies between concepts themselves will not reach the same performance for predicting concept mastery, although it approaches it within a few percentage points.

The results from this study indicate that the concept predictive accuracy of POKS remains between 1 – 3% below the BN's performance. However, the performance

advantage is reversed in favor of POKS for the question predictive accuracy. POKS appears to better predict observed item response outcome, but that advantage does not get reflected in concept prediction. Conversely, we note that the BN's concept prediction advantage does not reflect in its ability to predict item level outcome.

A simulation study that combines inferences from POKS and from the BN provides further insights. It shows that when the POKS item selection strategy is used, the BN's item prediction accuracy improves to even slightly higher levels than with POKS alone. However, that improvement does not transfer to concept prediction which remains close to the same level as without inferences from POKS. In the simulation where the BN item selection strategy is adapted take only item nodes into account, the results show that item prediction does improve, but not quite to the level obtained by combining the BN and POKS inferences, nor even to POKS item prediction alone. In other words, POKS does appear to better predict question outcome when the item selection strategy is optimized for reducing item level entropy as computed by POKS.

The POKS framework does not explicitly model concept dependencies but only item dependencies. This difference could explain why POKS does not reach the same concept predictive accuracy as the BN, which can take into account the information on other concepts mastery to determine the mastery of each concept. However, more experiments are required to confirm this explanation. On the other hand, the POKS framework yields a question accuracy performance systematically better than the BN framework. In another study, the POKS was compared with IRT framework and (Desmarais and Pu, 2005). Unlike the conventional use of IRT in CAT the industry, which usually has thousands of examinees in sample data set, the IRT model (2-PL) used in this study was limited to small sample data set. The results prove IRT is still a workable model, however quality of item parameter estimation may be compromised. Indeed, the specific POKS approach used in this comparison, using the same data as the IRT-2PL approach, provides similar accuracy for classifying students as master or non-master, but better item predictive accuracy.

These results confirm that knowledge items do have a structure among themselves and that it can be used to perform knowledge assessment, in accordance to the knowledge spaces theory (Doignon and Falmagne 1999). The higher performance of POKS compared to the BN one is also indicative that such structure can efficiently be modeled by partial order knowledge structures and that it can be used for knowledge assessment at the item level. We would still expect an AND/OR graph to yield yet a more accurate representation of knowledge structures and better predictive accuracy, but at least it appears that POKS can be effective when compared to BN and IRT at the item level. When using a simple weighted mean to assess concept mastery, as suggested in section 5.2, this is a significant finding. Although the POKS framework never reaches the performance of the BN to predict concept mastery, the fact that it approaches its performance within 1 – 3% in general and that it does well on item predictive accuracy is an encouraging step towards learned student models. The POKS framework is a relatively simple, computationally efficient, and a fully automated approach that can be applied with small data sets. As such, it offers many advantages over knowledge engineering approaches, namely the low effort required for model building and updating, its amenability to predict its accuracy under given circumstances such as sample size, and its avoidance of subjective biases and individual differences in expert modeling skill.

We also need to pay attention to the fact that that Vomlel's BN does not build links directly amongst question items themselves. This practice is typical of all BN used in the knowledge assessment and user modeling research literature. It also makes good sense since question items and assessment tests have a short life span and frequent updates. The knowledge engineering effort required to build a BN among test items would prove inefficient, unless the process can be fully automated as in POKS. Nevertheless, by not directly linking questions items among themselves, it is conceivable that the predictions miss valuable information that POKS exploited.

It is also possible that by relying on question items to infer concept mastery to, in turn, predict question mastery, the evidence propagated looses weights and gathers noise. This

could explain why direct links between question items themselves is more effective, in spite of the strong assumptions that used for building these links. Another, potentially more interesting hint at why the POKS did relatively well with question items lies in the structural properties of this domain. These properties are best understood by looking back at the theory of Knowledge Spaces we referred to in chapter 3. This theory is well known in mathematical psychology and it states that knowledge items are mastered in a constrained order.

Formally, Falmagne *et al.* (1990) argue that if the knowledge space of individual knowledge states is closed under union and intersection, then the set of all possible knowledge states can be represented by a directed acyclic graph (DAG).

In fact, Falmagne *et al.* (1990) show that the set of all knowledge states is closed under union only, not under intersection, and that an AND/OR graph is the proper structure. For our purpose, we make the assumption/approximation that it is closed under union and intersection and that a DAG is a proper representation of the ordering. This closure implies that, given a relation $A \Rightarrow B$, the absolute frequency of people who master a knowledge item A will necessarily be smaller than the frequency of B. This conclusion does not hold for the case of general BN.

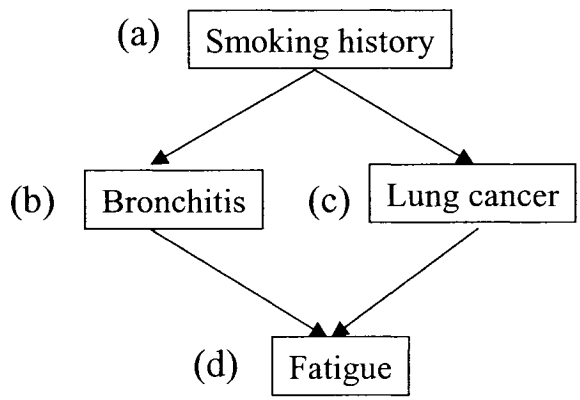


Figure 6.1. Example of lung cancer

For example, if we assume a structure (shown in figure 6.1.) as following (a BN taken from Neapolitan, 2004):

(a) Smoking history

(b) Bronchitis

(c) Lung cancer

(d) Fatigue

It is clear that smoking history (a) could be a much more frequent state than lung cancer (c) and bronchitis (b). It is also obvious that, whereas the occurrence lung cancer could decrease the probability of bronchitis by discounting that later cause as a plausible explanation for fatigue, discounting does not play a role in the case of knowledge structures (eg. observing (c) does not decrease the probability of (b); on the contrary, it could increase it).

In short, many interactions found in general BN do not occur in knowledge structures. We conjecture that this reduction in the space of possibilities that characterizes the domain we modeled in this experiment, namely the closure under union and intersection in knowledge spaces, warrants the use of strong independent assumptions in Bayesian modeling. It allows the modeling of the domain by a pairwise analysis of variable relations, thereby reducing considerably the computational complexity and the required size of the learning data set.

This last explanation is interesting because it links network structural properties (closure under union and intersection) to the level of assumption violation we can expect. However, we must emphasize that such explanation is speculative and not directly supported by empirical evidence from the current experiment. Further investigation is required to support such claim.

6.2 Conclusion

POKS offers a fully algorithmic means of building the model and updating item probabilities among themselves without requiring any knowledge engineering step. Indeed, the specific POKS approach uses the same data as the BN approach to provide similar accuracy. It shows that a graphical modeling approach such as POKS can be induced from a small amount of test data to perform relatively accurate examinee classification. This is an important feature from a practical perspective since the technique can benefit to a large number of application contexts.

The graphical modeling approaches such as POKS or as Bayesian networks are still in their infancy, and their potential benefit remains relatively unexplored. However, applications of CAT techniques to tutoring systems and to different learning environments are emerging.

The availability of simple and automated techniques that are both effective and efficient, relying on little data and allowing seamless updates of test content, will be critical to their success in commercial applications.

REFERENCES

ALMOND, R.G., MISLEVY, R.J. 1999. "Graphical models and computerized adaptive testing". *Applied Psychological Measurement*. 23: 3. 223-237.

ANASTASI, A. 1966. *Psychological testing*. New York: The Macmillan Company. 450p.

BAYES, T. 1763. "An essay towards solving a problem in the doctrine of chances". *Philosophical Transactions of the Royal Society*. 53. 370-418.

BEN-BASSAT, M. 1978. "Myopic policies in sequential classification". *IEEE Transactions on Computers*. 27: 2. 170-174.

BRETCH, B., JONES, M. 1988. "Student models: The genetic graph approach". *International Journal of Man Machine Studies*. 28: 5. 483-504.

BRUSILOVSKY, P., EKLUND, J., SCHWARZ, E. 1997. "Adaptive navigation support in educational hypermedia on the world wide web". *6th IFIP World Conference on Human-Computer Interaction (INTERACT97)*. Sydney, Australia. 278-285.

BRUSILOVSKY, P., EKLUND, J., SCHWARZ, E. 1998. "Web-based education for all: A tool for developing adaptive course-ware". *Seventh International World Wide Web Conference*. Brisbane, Australia. 291-300.

BRUSILOVSKY, P., SCHWARZ, E., WEBER, G. 1996. "A tool for developing adaptive electronic textbooks on WWW". *World Conference of the Web Society (WebNet'96)*. Boston, MA, USA. 64-69.

BURTON, R. 1982. "Diagnosing bugs in a simple procedural skill". *Intelligent tutoring systems*. Ed: Sleeman, D., Brown, J.S. London, New York: Academic. 345p.

CHENG, J., GREINER, R. 1999. "Comparing Bayesian network classifiers". *Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*. Ed: BLACKMOND LASKEY, K., PRADE, H. Pub: Morgan Kaufmann. 101-108.

CONATI, C., GERTNER, A., VANLEHN, K. 2002. "Using Bayesian networks to manage uncertainty in student modeling". *User Modeling and User-Adapted Interaction*. 12: 4. 371-417.

CONEJO, R., GUZMAN, E., MILLÁN, E., TRELLA, M., PÉREZ-DE-LA CRUZ, J.-L., RIOS, A. 2004. "SIETTE: A web-based tool for adaptive teaching". *International Journal of Artificial Intelligence in Education*. 14. 29-61.

DE FINETTI, B. 1926. *Teoria della Probabilità*. 1st ed. New York: John Wiley. 450p.

DESMARAIS, M.C., GIROUX, L., LAROCHELLE, S. 1993. "An Advice-Giving Interface Based on Plan-Recognition and User-Knowledge Assessment". *International Journal of Man-Machine Studies*. 39: 6. 901-924.

DESMARAIS M., MALUF, D.A., LIU, J. 1995. "User-Expertise Modeling with Empirically Derived Probabilistic Implication Networks". *User Modeling and User-Adapted Interaction*. 5: 3-4. 283-315.

DESMARAIS, M.C., MALUF, A., LIU, J. 1996. "User-expertise modeling with empirically derived probabilistic implication networks". *User Modeling and User-Adapted Interaction*. 5: 3-4. 283-315.

DESMARAIS, M.C., Pu, X. 2005a. "A Bayesian student model without hidden nodes and its comparison with item response theory". *Journal of Artificial Intelligence in Education*. (submitted).

DESMARAIS, M.C., PU, X. 2005b. "Computer adaptive testing: Comparison of a probabilistic network approach with item response theory". *10th International Conference on User Modeling (UM'2005)*. Edinburgh, UK.

DOIGNON, J.-P., FALMAGNE, J.-C. 1999. *Knowledge spaces*. Berlin, New York: Springer. 333p.

DOWLING, C.E., HOCKEMEYER, C. 2001. "Automata for the assessment of knowledge". *IEEE Transactions on Knowledge and Data Engineering*. 13:3. 451-461.

DUDA, R.O., HART, P.E., NILSSON, N.J. 1976. "Subjective Bayesian methods for rule based inference systems". *Readings in artificial intelligence. Ed:* Webber, B.L., Nilsson, N.J. Tioga Publishing, Palo Alto, CA. 192-199.

EGGEN, T.J.H.M. 2004. *Contributions to the Theory and Practice of Computerized Adaptive Testing*. 1st edition. Citogroep Arnhem, Netherlands: University of Twente. 226p.

FALMAGNE, J.-C., KOPPEN, M., VILLANO, M., DOIGNON, J.-P., JOHANNESEN, L. 1990. "Introduction to knowledge spaces: How to build test and search them". *Psychological Review*. 97. 201-224.

GAGNÉ, R. 1966. *The conditions of learning*. New York: Hold, Rinehart and Winston.

GIARRATANO, J.C., RILEY, G. 1989. *Expert systems: Principles and programming*. Boston: PWS-KENT Publishing. 597p.

GOLDSTEIN, I. 1982. The genetic graph: a representation for the evolution of procedural knowledge. *Intelligent tutoring systems*. Ed: Sleeman, D., Brown, J.S. London, New York: Academic. 51-77.

GOOD, I.J. 1983. *Good thinking*. Minneapolis, Minnesota: University of Minnesota Press. 313p.

HUGIN Inc. 2002. *Hugin Explorer*. Version 6.0. [Computer software]. Aalborg: Denmark. http://www.hugin.com.

JENSEN, F., KJÆRULFF, U.B., LANG, M., MADSEN, A.L. 2002. "Hugin - the tool for Bayesian networks and Influence diagrams". *First European Workshop on Probabilistic Graphical Models (PGM 2002)*. Ed: Gámez, J.A., Salmeron, A. 211-221.

KAMBOURI, M., KOPPEN, M., VILLANO, M., FALMAGNE, J.-C. 1994. "Knowledge assessment: tapping human expertise by the query routine". *International Journal of Human-Computer Studies*. 40: 1. 119-151.

KEYNES, J.M. 1921. *A treatise on probability*. Macmillan: London. 257p.

KHUWAJA, R., DESMARAIS, M.C., CHENG, R. 1996. "Intelligent guide: Combining user knowledge assessment with pedagogical guidance". *Third International Conference on Intelligent Tutoring Systems (ITS '96)*. London: Springer-Verlag. 225-233.

KOLMOGOROV, A.N. 1933. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. 1st edition. Berlin: Springer. 405p.

MAYO, M., MITROVIC, A., SURAWEERA, P., MARTIN, B. 2001. "Tutoring – Constraint-Based Tutors: A Success Story". *Lecture notes in Computer Science*. 2070. 931-940.

MILLÁN, E., TRELLA, M., PÉREZ-DE-LA-CRUZ, J.-L., CONEJO, R. 2000. "Using Bayesian networks in computerized adaptive tests". *Computers and education in the 21st century*. Ed: M. Ortega and J. Bravo. Belgium: Kluwer. 217-228.

NEAPOLITAN, R.E. 2004. *Learning bayesian networks*. 1st edition. New Jersey: Prentice Hall. 674 p.

SCHWARZ, E., BRUSILOVSKY, P., WEBER G. 1996. "World-wide intelligent textbooks". *World Conference on Educational Telecommunications (ED-TELECOM'96).* Boston, MA, USA. 302-307.

SPIRTES, P., GLYMOUR, C., SCHEINES, R. 1993. *Causation, Prediction, and Search.* New York: Springer-Verlag. 526p.

VANLEHN, K., LYNCH, C., SCHULZE, K., SHAPIRO, J., SHELBY, R., TAYLOR, L., TREACY, D., WEINSTEIN, A., WINTERSGILL, M. 2005. "The Andes physics tutoring system: Lessons learned". *International Journal of Artificial Intelligence and Education*. 15. 3-20.

VENABLES, W.N., SMITH, D.M., and the R Development Core Team. 2004. *An Introduction to R, notes on R: A Programming Environment for Data Analysis and Graphics*. Version 2.0.1. Bristol, UK: R Development Core Team. 90p.

VOMLEL, J. 2004. "Bayesian networks in educational testing". *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*. 12: Supplementary Issue 1. 83-100.

VON MISES R. 1928. *Probability, Statistics, and truth*. Allen and Unwin: London. 244p.

APPENDIX I. INFERENCE AND GRAPHICAL MODELING PACKAGES

The figure below provides a comparison of some software packages for graphical models. The columns have the following meanings.

Table I.1. Software packages for graphical models.

Name	Src	API	Cts	GUI	θ	G	$\sim$	U	UG	Free
Analytica	N	N	Y	W	N	N	N	Y	N	R
Bassist	C++	Y	Y	N	Y	N	Y	N	N	Y
Bayda	Java	-	Y	Y	Y	N	N	N	N	Y
BayesBuilder	N	N	N	Y	N	N	Y	N	N	R
B. Discoverer	N	N	D	Y	Y	Y	N	N	N	R
B-course	N	N	D	Y	Y	Y	N	N	N	Y
Bayonnet	Java	-	Y	Y	Y	N	N	N	N	Y
BN power constr.	N	W	N	Y	Y	CI	N	N	N	Y
BN Toolbox	Matlab	-	Y	N	Y	Y	Y	Y	N	Y
BN Toolkit	VBasic	-	N	Y	N	Y	N	N	N	Y
BucketElim	C++	-	N	N	N	N	N	N	N	Y
BUGS	N	N	Y	W	Y	N	Y	N	N	Y
Business Nav. 5	N	N	D	Y	Y	Y	N	N	N	R
CABeN	C	Y	N	N	N	N	Y	N	N	Y
CoCo+Xlisp	C/lisp	-	N	Y	Y	CI	N	N	Only	Y
CIspace	Java	N	N	Y	N	N	N	N	N	Y
Ergo	N	N	N	Y	N	N	N	N	N	R
FLoUE/BIFtoN	Java	-	N	N	N	N	N	N	N	Y
GDAGsim	C	-	Only	N	N	N	Y	N	N	Y
GMRFsim	C	-	Only	N	N	N	Y	N	Only	Y
GeNIe/SMILE	N	WU	N	W	N	N	Y	Y	N	Y
Hydra	Java	-	Y	N	Y	N	Y	N	Y	Y
Hugin Expert	N	Y	Y	W	Y	CI	Y	Y	Y	R
Ideal	Lisp	-	N	Y	N	N	N	Y	N	Y
Java Bayes	Java	-	N	Y	N	N	N	Y	N	Y
MIM	N	N	Y	Y	Y	Y	N	N	Y	R
MSBNx	N	Y	N	W	N	N	N	Y	N	R
Netica	N	WUM	Y	W	Y	N	Y	Y	N	R
Pulcinella	Lisp	-	N	Y	N	N	N	N	N	Y
RISO	Java	-	Y	Y	N	N	N	N	N	Y
Tetrad	N	N	Y	N	Y	CI	N	N	Y	Y
Web Weaver	Java	-	N	Y	N	N	N	Y	N	Y
WinMine	N	N	Y	Y	Y	Y	N	N	Y	R
XBAIES 2.0	N	N	N	Y	Y	N	N	Y	Y	Y

- Src: Is source code available

- API: Is an application programmable interface available, or must the program be used as a standalone black box? (-= source is included, so no API is needed; W=windows, U=Unix, M=Mac)
- Cts: Does the package supports continuous random variable (D means it discretizes them)
- GUI: Does the package have a graphical user interface?
- Θ: Does the package learn parameters? (N = no, Y = uses search and score, CI = uses conditional independence tests)
- ~:Does the package support sampling?
- U: Does the package support utility/action nodes (i.e., decision diagrams)?
- UG: Does the package support undirected graphs?
- Free: Is the package freely available? (Y=yes, N=no, R=restricted (commercial but free for non-commercial use, and/or free version has limited functionality)

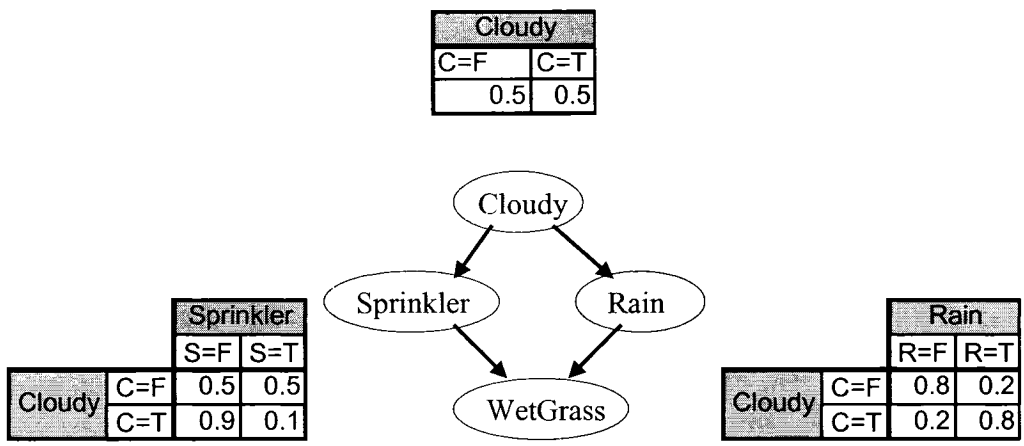
Bayesian Network Java, Matlab and C Tools (Free and Commercial)

1. BNJ (http://bnj.sourceforge.net/)
2. Java Bayes (http://www-2.cs.cmu.edu/~javaBayes/)
3. Bayes Builder (http://www.snn.kun.nl/nijmegen/index.php3?page=31)
4. Bayesia Lab (http://www.Bayesia.com/)
5. Bayesware (http://www.Bayesware.com/)
6. Bayes Net Toolbox (BNT) for Matlab (http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html)
7. GDAGSIM in C (http://www.staff.ncl.ac.uk/d.j.wilkinson/software/gdagsim/)
8. GMRF Sim in C (http://www.math.ntnu.no/~hrue/GMRFsim/)
9. C++ Version of BNT (http://www.intel.com/research/mrl/pnl/)
10. WinMine (http://research.microsoft.com/~dmax/WinMine/tooldoc.htm)
11. BUGS (http://www.mrc-bsu.cam.ac.uk/bugs/welcome.shtml)
12. Bayesian Logistic Regression Software (http://www.stat.rutgers.edu/~madigan/BBR/)
13. jNBC (http://jbnc.sourceforge.net/)
14. Netica (http://www.norsys.com/)
15. CleverSet (http://www.cleverset.com/)
16. Knowledge industries (http://www.kic.com/products.html)
17. Pulcinella (http://iridia.ulb.ac.be/pulcinella/Welcome.html)
18. BAYDA (http://www.cs.helsinki.fi/research/cosco/Projects/NONE/SW/)
19. CISpage (http://www.cs.ubc.ca/labs/lci/CIspace/version2/Bayes.html)

20. DecisionQ (http://www.decisionq.com/)
21. Belief Network Power Constructor (http://www.cs.ualberta.ca/~jcheng/bnpc.htm)
22. XBAIES (http://www.staff.city.ac.uk/~rgc/webpages/xbpage.html)
23. Webweaver III (http://snowhite.cis.uoguelph.ca/faculty_info/yxiang/ww3/)
24. Ergo (http://www.noeticsystems.com/ergo/)
25. Bayes Line – C++ (http://Bayesline.sourceforge.net/)
26. HUGIN * (http://www.hugin.com/Products_Services/Products/)

APPENDIX II. BNT EXAMPLE

Here we provide a simple example to explain how a DAG and a Bayesian network is constructed in BNT Matlab Toolbox and how inferences are conducted. The structure and the CPD tables are presented in the following figure.



Cloudy	
C=F	C=T
0.5	0.5

		Sprinkler	
		S=F	S=T
Cloudy	C=F	0.5	0.5
	C=T	0.9	0.1

		Rain	
		R=F	R=T
Cloudy	C=F	0.8	0.2
	C=T	0.2	0.8

		Wet Grass	
Sprinkler	Rain	W=F	W=T
S=F	R=F	1	0
S=T	R=F	0.1	0.9
S=F	R=T	0.1	0.9
S=T	R=T	0.01	0.99

Figure II.1. Bayes inference for a rainy day

To specify this directed acyclic graph (DAG), we create an adjacency matrix:

```
N = 4; % number of the nodes
dag = zeros(N,N);
C = 1; S = 2; R = 3; W = 4; %Naming conventions for nodes
dag(C,[R S]) = 1; % Creating the arcs
dag(R,W) = 1;
dag(S,W)=1;
```

Nodes are numbered in topological order as:

```
Cloudy = 1, Sprinkler = 2, Rain = 3, WetGrass = 4.
discrete_nodes = 1:N;
node_sizes = 2*ones(1,N);
bnet = mk_bnet(dag, node_sizes, 'discrete', discrete_nodes);
```

Now it is time to represents the parameters by CPD objects (CPD = Conditional Probability Distribution), which define the probability distribution of a node given its parents.

```
bnet.CPD{C} = tabular_CPD(bnet, C, [0.5 0.5]);
bnet.CPD{R} = tabular_CPD(bnet, R, [0.8 0.2 0.2 0.8]);
bnet.CPD{S} = tabular_CPD(bnet, S, [0.5 0.9 0.5 0.1]);
bnet.CPD{W} = tabular_CPD(bnet, W, [1 0.1 0.1 0.01 0 0.9 0.9 0.99]);
```

Having created the BN, we can now use it for inference. We will use the junction tree engine, which is the mother of all exact inference algorithms:

```
engine = jtree_inf_engine(bnet);
```

Suppose we want to compute the probability that the sprinker was on given that the grass is wet. The evidence consists of the fact that W=2. All the other nodes are unobserved. We can specify this as follows.

```
evidence = cell(1,N);
evidence{W} = 2;
```

We are now ready to add the evidence to the engine.

```
[engine, loglik] = enter_evidence(engine, evidence);
```

Finally, we can compute p=P(S=2|W=2) as follows.

```
marg = marginal_nodes(engine, S);
marg.T
ans =
    0.57024
    0.42976
p = marg.T(2);
```

We see that p = 0.4298.

Now let us add the evidence that it was raining, and see what difference it makes.

```
evidence{R} = 2;
[engine, loglik] = enter_evidence(engine, evidence);
marg = marginal_nodes(engine, S);
p = marg.T(2);
```

We find that p = P(S=2|W=2,R=2) = 0.1945, which is lower than before, because the rain can ``explain away'' the fact that the grass is wet.

APPENDIX III. INFORMATION ENTROPY

Entropy is a concept in thermodynamics, statistical mechanics and information theory. The concepts of information and entropy have deep links with one another, although it took many years for the development of the theories of statistical mechanics and information theory to make this apparent. This appendix is about information entropy and the formulation of entropy.

The basic concept of entropy in information theory has to do with how much randomness there is in a signal or random event. An alternative way to look at this is to talk about how much information is carried by the signal (we use signal here because shanon has developed his theory for signal processing applications).

As an example consider some English text, encoded as a string of letters, spaces and punctuation (so our signal is a string of characters). Since some characters are not very likely (e.g. 'z') while others are very common (e.g. 'e') the string of characters is not really as random as it might be. On the other hand, since we cannot predict what the next character will be, it does have some 'randomness'. Entropy is a measure of this randomness, suggested by Claude E. Shannon in his 1948 paper A Mathematical Theory of Communication.

Shannon derives his definition of entropy from the assumptions that:

- The measure should be proportional (continuous) meaning changing the value of one of the probabilities by a very small amount should only change the entropy by a small amount.
- If all the outcomes (letters in the example above) are equally likely then increasing the number of letters should always increase the entropy.

- We should be able to make the choice (in our example of a letter) in two steps, in which case the entropy of the final result should be a weighted sum of the entropies of the two steps.

He defines entropy in terms of a discrete random event x, with possible states $1..n$ as:

$$H(x) = \sum_{i=1}^{n} p(i) \log_2 (\frac{1}{p(i)}) = -\sum_{i=1}^{n} p(i) \log_2 p(i)$$

That is, the entropy of the event x is the sum, over all possible outcomes i of x, of the product of the probability of outcome i times the log of the probability of I.